



Technische
Universität
Braunschweig



Deutsches Zentrum
für Luft- und Raumfahrt

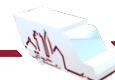
Masterarbeit

Entwicklung eines Serious Games zur Konzeptvermittlung von Mobility-on-demand Systemen

Technische Universität Carolo-Wilhelmina zu Braunschweig
Institut für Psychologie, Abteilung Verkehrspsychologie

Niels Kowala Matrikelnummer: 4817604

Erstgutachter : Prof. Dr. Mark Vollrath
Zweitgutachter : Prof. Dr.-Ing. Harald Michalik
Betreuerin : M.Sc. Alexandra König
Eingereicht am : 20. Juni 2018



Erklärung

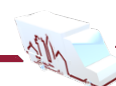
Hiermit versichere ich, die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt zu haben. Die verwendete Literatur und sonstige Hilfsmittel sind vollständig angegeben.

Braunschweig, 17. Juni 2018

Inhaltsverzeichnis

Erklärung	II
Kurzfassung	V
Abbildungsverzeichnis	VI
1 Einleitung	1
1.1 Problemstellung	1
1.2 Aufbau der Arbeit	2
1.3 Serious Games	2
1.4 Serious Game B.u.S.	3
1.4.1 Reallabor Schorndorf	3
1.4.2 Ziele des Serious Games	6
1.4.3 Bisheriger Stand	6
1.5 Bestehende ähnliche Spielkonzepte	7
1.6 Analyse der Spielkonzepte	11
2 Vorgehensmodell für die Softwareentwicklung	12
2.1 Grundlagen Vorgehensmodelle	12
2.2 Iterative Entwicklung	13
2.2.1 Spiralmodell nach Boehm (1988)	14
2.2.2 Playtesting-Modell	15
2.2.3 Game Development Life Cycles	17
2.2.4 Vorgehensmodell für Serious Games	18
2.3 Auswahl eines Vorgehensmodells	19
2.3.1 Anforderungen an ein Vorgehensmodell für das Projekt B.u.S.	19
2.3.2 Analyse der vorgestellten Vorgehensmodelle	20
2.3.3 Angepasstes iteratives Vorgehensmodell zur Entwicklung und Evaluation von Serious Games	20
3 Iterative Entwicklung	23
3.1 Iteration 1 - Konzeptphase	23
3.1.1 Methoden	23
3.1.2 Didaktische Anforderungen	27
3.1.3 Spielmodell	27
3.2 Iteration 2 - Fokusgruppengespräch	28
3.2.1 Methode	29
3.2.2 Ziel der Fokusgruppe	30

3.2.3	Teilnehmer	30
3.2.4	Durchführung	30
3.2.5	Vorgestellte Konzepte	32
3.2.6	Auswertung der entwickelten Konzepte	35
3.2.7	Meinungen	36
3.2.8	Bewertung der Fokusgruppenmethode	37
3.3	Iteration 3 - Heuristische Evaluation	38
3.3.1	Methode	38
3.3.2	Teilnehmer	42
3.3.3	Durchführung und Testsetting	43
3.3.4	Gefundene Probleme	44
3.3.5	Auswertung der Heuristiken und der Schweregrade	48
3.3.6	Maßnahmen	48
3.4	Iteration 4 - Thinking Aloud Test	50
3.4.1	Evaluationsdesign und Durchführung	50
3.4.2	Beobachtungen	51
3.4.3	Auswertung	52
3.5	Iteration 5 - Softwaretests	55
4	Projektmanagement und Konzeptrealisierung	56
4.1	Projektcontrolling	56
4.2	Anforderungsmanagement	56
4.3	Software-Framework	58
4.4	Softwarearchitektur	59
4.5	Spielablauf	60
4.6	Spielelemente	60
4.7	Spiellevel	65
5	Diskussion	67
5.1	Reflektion der Ergebnisse	67
5.1.1	Allgemeine Beobachtungen bei den Evaluationen	67
5.1.2	Heuristische Evaluation	68
5.1.3	Thinking Aloud	69
5.2	Restriktionen	70
5.3	Ausblick und Empfehlungen	71
A	Anhang	72
	Literaturverzeichnis	73



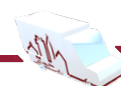
Kurzfassung

Die vorliegende Masterarbeit untersucht die Konzeptionierung und iterative Entwicklung eines Serious Games. Das Serious Game (B.u.S.) soll das Konzept eines Mobility-On-Demand Systems (auch genannt Rufbus) vermitteln. Die Entwicklung des Serious Games erfolgt in Zusammenarbeit mit dem Deutschen Zentrum für Luft- und Raumfahrt e. V. (DLR) im Rahmen des Projekts Reallabor Schorndorf. Für die Software-Entwicklung werden iterative Vorgehensmodelle analysiert und ein angepasstes iteratives Vorgehensmodell wird vorgestellt. Dieses iterative Vorgehensmodell beginnt mit der Konzeptentwicklung und berücksichtigt neben einem Fokusgruppengespräch zur Konzeptfindung Usability Tests zur Untersuchung der Benutzerfreundlichkeit. Als Usability Tests werden in dieser Arbeit der Thinking Aloud Test und die Heuristische Evaluation vorgestellt. Die Konzeptentwicklung erfolgt mit Hilfe des Mechanics-Dynamics-Aesthetics (MDA) Frameworks. Alle Ergebnisse der durchgeführten Usability-Evaluationen sind in einen Prototypen des Serious Games eingeflossen, der Prototyp wurde mit Hilfe der Unity-Engine entwickelt. Abschließend stellt diese Arbeit die umgesetzten didaktischen Elemente des Serious Games und den Aufbau der entstandenen Software vor.

Abbildungsverzeichnis

1.1	Logo des Projekts (Stadt Schorndorf, 2017)	4
1.2	Kleinbusse mit der Lackierung des Projekts (Stadt Schorndorf, 2017)	5
1.3	Mögliche Haltestellen des Bedarfsbusses in Schorndorf (Stadt Schorndorf, 2017)	5
1.4	Startbildschirm zu dem Spielkonzept (Wegener, 2017)	7
1.5	Spielszene (Screenshot) aus dem Spiel MiniMetro	8
1.6	Spielszene aus dem Computerspiel Cities in Motion 2 (Deppe, 2013)	9
1.7	Spielszene aus dem Serious Game ProtoWorld (GaPSlabs, o.D.)	10
1.8	Serious Game BusMeister (Nash, o.D.)	11
2.1	Iteratives Vorgehensmodell mit fünf Phasen (Kleuker, 2013, S. 31)	14
2.2	Spiralmodell nach Boehm (Boehm, 1988)	15
2.3	Playtesting-Modell (Fullerton, 2008, S. 249)	16
2.4	Beispiel eines Game Development Life Cycles (Ramadan & Widayani, 2013)	18
2.5	Vorgehensmodell für Serious Games (Strahringer & Leyh, 2017, S. 20)	19
2.6	Angepasstes iteratives Entwicklungsmodell (eigene Abbildung)	22
3.1	MDA-Framework nach Hunicke u. a. (2004)	26
3.2	Ablaufschema eines Fokusgruppenprojekts (Salzburg Research Forschungsge- sellschaft, o.D.)	29
3.3	Gezeigtes Konzept beim Fokusgruppengespräch	32
3.4	Konzeptskizze von „Schlag den Linienverkehr“	33
3.5	Konzeptskizze von „Transport Tycoon“	34
3.6	Anzahl der gefunden Probleme nach Nielsen (2000)	39
3.7	Entwicklungsstand zum Zeitpunkt der Heuristischen Evaluation	43
3.8	Testsetting bei der heuristischen Evaluation	44
3.9	Entwicklungsstand: Ausschnitt aus dem Serious Game	51
3.10	Nennungen der Probleme aufgeteilt nach Testern (T1 - T5)	52
4.1	Kanban-Board in Trello	58
4.2	Softwarearchitektur von B.u.S.	59

4.3	Aktivitätsdiagramm des Spielablaufs	60
4.4	Rufbus- und Linienbushaltestellen im Spiel	61
4.5	Busmodelle im Spiel	62
4.6	Fahrgäste im Spiel	63
4.7	Infoleiste aus Level 5 mit Darstellung der Umweltfreundlichkeit	63
4.8	Einführungstext aus dem ersten Level	64
4.9	Reflektionstext nach dem Erreichen eines Spielziels	65
4.10	Planung der Spiellevel	66
5.1	Nennungen der Probleme bei den Evaluationen	68



1 Einleitung

„Wir behalten von unsern Studien am Ende doch nur das, was wir praktisch anwenden.“

Johann Wolfgang von Goethe (1749 - 1832)

Mit dem oben genannten Zitat wollte Goethe ausdrücken, dass Lerninhalte besser gelernt werden, wenn diese praktisch angewendet werden (Eckermann, 1836). Ein digitales Serious Game ist eine moderne Art das Lernen mit einer praktischen Anwendung in Form eines Spiels zu verknüpfen.

1.1 Problemstellung

Diese Arbeit untersucht die iterative Entwicklung eines Serious Games für ein Rufbus-Konzept. Das Serious Game soll ein Spiel sein, das die Funktionsweise des Rufbusses erläutert und die Akzeptanz dessen steigert. Das Spiel soll dabei die Einführung des neuen Bussystems begleiten. Da das System ein grundlegend geändertes Konzept vorsieht, wird angenommen, dass die Benutzerakzeptanz und der Benutzungswille für den Rufbus erst aufgebaut werden muss. Ein Serious Game ist ein interaktiver moderner Ansatz für diese Ziele. Mit dem Serious Game sollen potentielle Fahrgäste bereits vor der Benutzung das Konzept und den Rufbus auf eine spielerische Weise kennen lernen.

Auf Grund der kurzen Entwicklungszeit muss die Entwicklung des Serious Games agil erfolgen. Dennoch ist eine möglichst gute Usability für das Serious Game gewünscht. Um diese zu gewährleisten, soll ein iterativer agiler Entwicklungsansatz untersucht werden, der mehrere Evaluationen vorsieht.

Die Didaktik in Serious Games und das Lernen mit Serious Games sind Randthemen in dieser Arbeit und werden weniger betrachtet.

1.2 Aufbau der Arbeit

Das erste Kapitel beinhaltet eine Einleitung zu dem Thema der Arbeit und erläutert das zu dem Serious Game zugehörige Projekt „Reallabor Schorndorf“. Anschließend wird der bisherige Projektstand des Serious Games beschrieben. Dieses Kapitel gibt nach einer Begriffsdefinition eine Übersicht über ähnliche bestehende Spielkonzepte und analysiert diese.

Das zweite Kapitel beschreibt das verwendete Vorgehensmodell, geht auf die Grundlagen und die wichtigsten iterativen Vorgehensmodelle ein. Nach der Vorstellung der Vorgehensmodelle wird ein Modell für die weitere Benutzung ausgesucht und ein angepasstes Vorgehensmodell vorgestellt.

Der Verlauf des dritten Kapitels orientiert sich an den Iterationen des zuvor vorgestellten Vorgehensmodells. Es beschreibt die Iterationen mit den durchgeführten Evaluationen.

Der vierte Teil der vorliegenden Arbeit beschreibt das Projektmanagement, die Architektur der Software und geht auf die Realisierungen der Spielelemente ein.

Der fünfte Teil der Arbeit reflektiert die Ergebnisse und beschreibt die Restriktionen für den Einsatz des vorgestellten Vorgehensmodells.

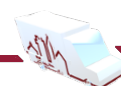
1.3 Serious Games

Die erste Erwähnung und Definition des Begriffs Serious Game fand durch Abt (1970) statt. Dieser beschrieb Serious Games als Spiele mit eindeutigem Bildungszweck, die nicht nur der Unterhaltung dienen. Vor den Videospielen fand der Begriff vor allem für wissensvermittelnde Brett- und Kartenspiele Verwendung (Abt, 1970).

Sawyer & Rejeski (2002) prägten den Begriff für softwarebasierte Spiele mit realitätsbasierten Simulationen zur Lösung definierter Aufgaben und berücksichtigten damit die Digitalisierung der Spiele.

Zyda (2005) beschrieb Serious Games allgemeiner als Videogames, die den Spieler¹ unterhalten sollen und gleichzeitig weiterbilden oder instruieren. Sie müssen nicht unbedingt eine Simulation enthalten und können auch ohne definierte Lernaufgaben Wissen vermitteln.

¹Aus Gründen der Lesbarkeit wird in der vorliegenden Arbeit die männliche Form stets gleichberechtigt für die weibliche verwendet.



Zusammengefasst beinhalten alle Aussagen, dass Serious Games nicht nur zur reinen Unterhaltung dienen, sondern auch die Absicht haben Lern- und Übungseffekte zu erzielen. Sie können Informationen, theoretisches Wissen und erfahrungsbasiertes Wissen vermitteln (Strahringer & Leyh, 2017).

In der deutschen Sprache werden für Serious Games auch die Begriffe *ernsthafte Spiele* oder *Lernspiele* verwendet (Graf, 2008). Serious Games können zur Gamification genutzt werden. Mit dem Begriff Gamification werden Systeme beschrieben, die einzelne spielerische Elemente besitzen und in produktiven Bereichen eingesetzt werden. Serious Games werden gezielt auch außerhalb einer produktiven Tätigkeit und mit anderen Erwartungen eingesetzt (Graf, 2008).

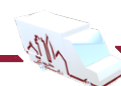
1.4 Serious Game B.u.S.

Das mit dieser Arbeit eingeführte Serious Game wird unter dem Titel „Bürger unterrichten durch Spiele“ (B.u.S.) beim Deutschen Zentrum für Luft- und Raumfahrt (DLR) entwickelt. Nachfolgend wird die Abkürzung B.u.S. für die Beschreibung des Serious Games genutzt.

Das Serious Game soll die Einführung eines Rufbusses in der Stadt Schorndorf unterstützen. Das Projekt dazu trägt den Namen Reallabor Schorndorf. Der in dem Spiel beschriebene Mobility-On-Demand-Bus wird in der Fachliteratur (BMVBS, 2009) auch als Rufbus oder Bedarfsbus bezeichnet. Die Begriffe haben nachfolgend die gleiche Bedeutung. Die verschiedenen Begriffe sind durch die historische Entwicklung entstanden. Die beiden nachfolgenden Unterkapitel erläutern das Projekt und die Ziele des Serious Games.

1.4.1 Reallabor Schorndorf

Das Projekt „Reallabor Schorndorf: Zukunftsweisender Öffentlicher Verkehr – Bürgerorientierte Optimierung der Leistungsfähigkeit, Effizienz und Attraktivität im Nahverkehr“ soll ein innovatives bedarfsorientiertes Buskonzept entwickeln und untersuchen. Das Logo des Projekts ist in Abbildung 1.1 dargestellt. Das Ministerium für Wissenschaft, Forschung und Kunst des Landes Baden-Württemberg fördert das Projekt als eines von sieben Projekten, die partizipative Verfahren zur Unterstützung von Transformationsprozessen erproben. An dem Projekt beteiligt sind neben dem Deutschen Zentrums für Luft- und Raumfahrt das Zentrum für Interdisziplinäre Risiko- und Innovationsforschung der Universität Stuttgart, die Hochschule Esslingen, der Verkehrs- und Tarifverbund Stuttgart, der lokale Busbetreiber



Knauss Linienbusse und die Stadt Schorndorf. Ziel des Projekts ist es, ein Konzept für ein Bedarfsbus für Zeiten mit schwächerer Nachfrage zu entwickeln. Der Bedarfsbus soll über eine App, über die Homepage oder mit Hilfe eines Telefonanrufs individuell abrufbar sein. Während des Versuchszeitraums wird der Bus jeweils an den Wochenenden von Freitagnachmittag bis Sonntagabend eingesetzt. Der Pilotbetrieb startete am 10. März 2018.



Abbildung 1.1: Logo des Projekts (Stadt Schorndorf, 2017)

Der Anrufbus bedient den sogenannten bedarfsgesteuerten Flächenbetrieb. Das heißt, dass der Bus feste Haltestellen bedient, die Abfahrzeiten und die Route sind allerdings flexibel und werden an dem Bedarf angepasst. Der Fahrplan eines einzelnen Busses wird mit Hilfe einer Software kontinuierlich angepasst und den Busfahrern mitgeteilt. Der Bedarf ist über die vorherige Buchung eines Busses bekannt. Die Busse und fahren ausschließlich auf Bestellung. Im Gegensatz zu einem Taxi gibt es Fahrtenbündlungen. Das heißt, die Fahrgäste müssen einen kleinen Umweg bei mehreren Fahrgästen in Kauf nehmen. Die aktuellen Fahrkartenpreise für die Anrufbusse orientieren sich an den Preisen für den Linienverkehr.

Für den Bedarfsbus werden Kleinbusse eingesetzt, die effizienter und ressourcenschonender sind als Linienbusse. Sie sollen neben der Umwelt auch das Verkehrsaufkommen entlasten (Stadt Schorndorf, 2017). Die eingesetzten Kleinbusse sind vom Typ Mercedes Sprinter, wie in Abbildung 1.2 zu sehen ist.

Eingesetzt werden zwei Busse: Ein Hybrid-Bus mit 7 Sitzplätzen und ein barrierefreier Bus mit 10 Sitzplätzen. Das Konzept sieht eine deutlich höhere Haltestellendichte im Vergleich zum Linienbus vor. Die Haltestellendichte wird erreicht durch virtuelle Haltestellen, die nur online und auf Karten einsehbar sind (Stadt Schorndorf, 2017). Die Abbildung 1.3 zeigt die bestehenden Linienbus-Haltestellen und die neuen Rufbus-Haltestellen.

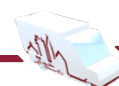




Abbildung 1.2: Kleinbusse mit der Lackierung des Projekts (Stadt Schorndorf, 2017)

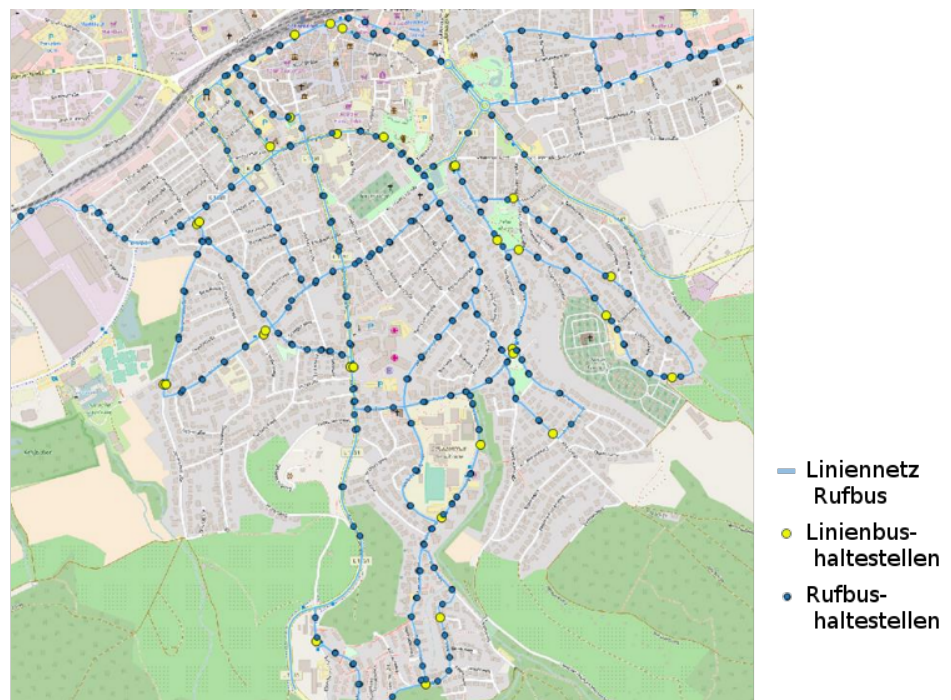
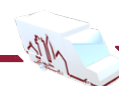


Abbildung 1.3: Mögliche Haltestellen des Bedarfsbusses in Schorndorf (Stadt Schorndorf, 2017)



1.4.2 Ziele des Serious Games

Mit Hilfe des Serious Games B.u.S. sollen sich die Bürger der Stadt Schorndorf aktiv mit dem neuen Bedienkonzept des Busses auseinander setzen. Das Spiel wird mit den folgenden drei Hauptzielen entwickelt (siehe auch König u. a., 2017):

- Wissensvermittlung zum Thema bedarfsgesteuerte Bedienung und Abgrenzung zu liniengebundenen Systemen
- Verbesserung der Einstellung der Spieler gegenüber des neuen Konzepts durch Aufzeigen der Vorteile
- Steigerung der Nutzungsintention bei möglichen Fahrgästen

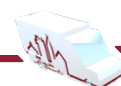
Das Serious Games B.u.S. soll mit diesen Zielen das neue Buskonzept erklären um die Akzeptanz des Buskonzepts zu steigern und die Vorteile des Konzepts zu zeigen. Neben der Erläuterung des neuen Buskonzepts sollen die Unterschiede zum bekannten Linienbusverkehr deutlich werden.

Das Spiel ist als eine Art Puzzlespiel geplant. Der Begriff Puzzle steht für ein regelbasiertes Spiel, in dem der Spieler eine Lösung finden muss (Fullerton, 2008). Die Spieler sollen sich durch den Spielansatz freiwillig mit dem Thema auseinander setzen. Die Spieler sind potentielle Nutzer des Buskonzepts in Schorndorf. Die Bedingungen des Spiels sollen sich an den realen Bedingungen orientieren. Das heißt, dass das Spiel das Straßennetz von Schorndorf nutzen sollte und die Orte der Haltestellen im Spiel sich an den realen Orten (siehe Abbildung 1.3) orientieren sollen.

1.4.3 Bisheriger Stand

Beschreibung der vorherigen Entwicklung

Zu der Design- und Konzeptentwicklung des Serious Games wurden bereits zwei Masterarbeiten verfasst, in denen Konzepte und Spielelemente entwickelt wurden. Die erste Masterarbeit ist eine theoretische Abhandlung für das Design und die Benutzeroberfläche (Pelz, 2017). Die Arbeit stellt ein virtuelles Stadtmodell und ein Design für einen virtuellen Assistenten vor. Die zweite Masterarbeit umfasst die Entwicklung einer theoretischen Assistenten-KI (Wegener, 2017).



Das erste Konzept für das Serious Game sieht vor, die Stadt in verschiedene Bereiche einzuteilen. Die Spielzeit ist ein virtueller Tag, der ebenfalls in verschiedenen Blöcke aufgeteilt ist. Es gibt verschiedene Personengruppen mit unterschiedlichen Mobilitätsbedarfen. Der Spieler übernimmt die Rolle des Verkehrsplaners, platziert Haltestellen und kauft Busse ein. Anschließend kann der Spieler die Busse mit Verkehrslinien verknüpfen (Wegener, 2017).

Analyse des bisherigen Stands

Bei Gesprächen mit Human-Factors-Experten vom DLR hat sich das Konzept als sehr komplex und umfangreich herausgestellt. Eine implementierte Spiellogik ist nur von einigen wenigen Spielelementen vorhanden. Allerdings kann von der ersten Masterarbeit das Straßennetz übernommen werden, von der zweiten Masterarbeit kann die Dialogführung mit Hilfe des Assistenten übernommen werden. Der letzte Stand der Umsetzung ist in Abbildung 1.4 visualisiert. Die Abbildung zeigt den Assistenten mit dem Straßennetz der Stadt im Hintergrund.

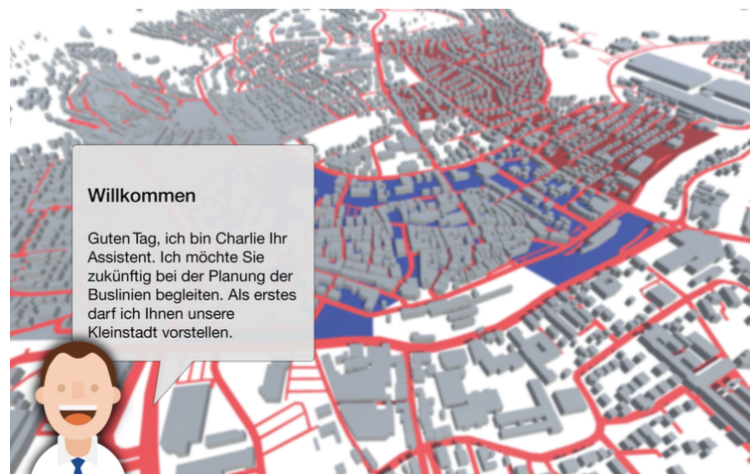
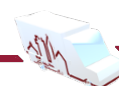


Abbildung 1.4: Startbildschirm zu dem Spielkonzept (Wegener, 2017)

1.5 Bestehende ähnliche Spielkonzepte

Dieses Kapitel beschreibt den aktuellen Stand der Entwicklung von relevanten Spielkonzepten. Die nachfolgenden Spiele und Serious Games sind relevant, da sie ähnliche Spielkonzepte wie B.u.S. besitzen. Es werden die beiden Spiele Mini Metro, Cities in Motion und die beiden Serious Games Proto-World und BusMeister beschrieben.



Mini Metro

Das Mobile-Game Mini Metro ist ein Puzzle-Spiel mit dem Hauptziel der Unterhaltung. Es verfolgt kein explizites pädagogisches Ziel und ist damit kein Serious Game (Rath, 2017). Dennoch zeigt es Ansätze eines Serious Games, denn es erläutert die Komplexität und Probleme eines Metro-Netzes. Eine Spielszene mitsamt der minimalistischen Benutzeroberfläche zeigt die Abbildung 1.5.

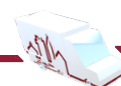
Das Spiel basiert auf einfachen abstrakten geometrischen Formen, die Metrostationen, Passagiere, Metrobahnen und Metrolinien darstellen sollen. Ziel des Spiels ist es möglichst viele Passagiere zu transportieren. Dazu muss der Spieler die vorhanden Stationen zu Linien verbinden und Metrobahnen einsetzen. Sowie die Anzahl und Varianten der Stationen als auch die Anzahl der Passagiere wächst mit der Spielzeit. Das Spielende wird durch die Überlastung (zu viele wartenden Passagiere) einer Station ausgelöst. Grundlage für die Level sind reale Städte, die abstrahiert wurden. Mini Metro ist ein gutes Vorbild für eine intuitive Benutzeroberfläche und für eine minimalistische, aber dennoch nachhaltige Spielmechanik.



Abbildung 1.5: Spielszene (Screenshot) aus dem Spiel MiniMetro

Cities in Motion 1/2

Cities in Motion ist kein Serious Game hat aber ein relevantes Konzept. Der Spieler muss in dem Spiel ein Nahverkehrsnetz in einer existierenden Stadt aufbauen (Deppe, 2013). Ziel des Spiels ist es, durch ein profitables Verkehrsnetz möglichst viel Geld zu verdienen. Im



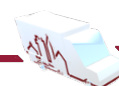
Gegensatz zu Mini Metro basiert Cities in Motion auf einer realistischen Umsetzung und Simulation (siehe Abbildung 1.6).



Abbildung 1.6: Spielszene aus dem Computerspiel Cities in Motion 2 (Deppe, 2013)

ProtoWorld

ProtoWorld ist eine Simulation von Fahrgästen und Buslinien im operativen Betrieb. Das Ziel von ProtoWorld ist, das Bewusstsein von Stakeholdern für den ÖPNV zu steigern (GaPSlabs, o.D.). Für die Simulationen können reale Kartendaten importiert werden, diese Kartendaten können zum Beispiel von Open-StreetMap stammen. Die Simulation basiert ebenfalls wie B.u.S. auf dem Unity-Framework. Die Abbildung 1.7 zeigt einen Ausschnitt einer Simulation mit ProtoWorld. Dargestellt ist die Simulation eines Busbahnhofs in Delft.



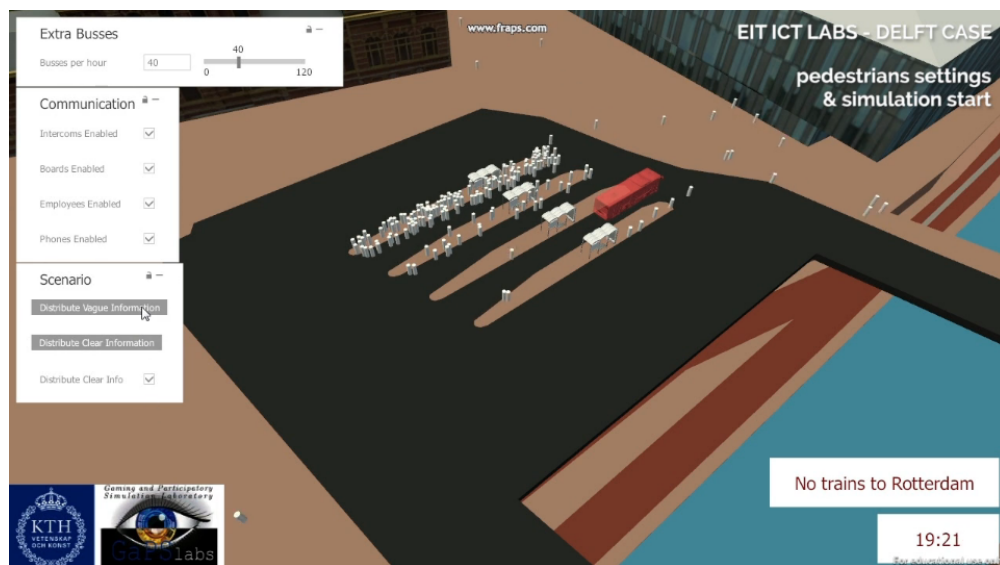


Abbildung 1.7: Spielszene aus dem Serious Game ProtoWorld (GaPSlabs, o.D.)

BusMeister

BusMeister ist eine internetbasierte Plattform zur Einbindung der Bevölkerung in die Planung des ÖPNVs (Nash, o.D.). Dazu unterrichtet die Plattform die Bevölkerung über das Thema nachhaltiger Nahverkehr. Zusätzlich gibt es ein Internetforum für Interaktionen und ein Serious Game für die Wissensvermittlung.

Das BusMeister-Spiel hat das Ziel, lokale Verbesserungsmöglichkeiten für den ÖPNV zu unterrichten (Nash, o.D.). Als zweites Ziel soll das Spiel die Bekanntheit der Website und des Forums steigern. Das Spiel basiert auf einer Schrägprojektion einer einfachen Straße, wie in Abbildung 1.8 zu sehen ist. Der Spieler kann in dem Spiel Ampeln, Bushaltestellen und Busspuren anordnen und Busgrößen auswählen. Eine anschließende Simulation mit anderen Verkehrsteilnehmern berechnet die Zufriedenheit der Anwohner. Der Einsatz der passenden Verkehrselemente entscheidet über die Zufriedenheit und damit auch über den Spielerfolg.

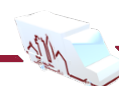




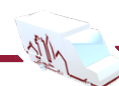
Abbildung 1.8: Serious Game BusMeister (Nash, o.D.)

1.6 Analyse der Spielkonzepte

Die vorgestellten Spielkonzepte haben unterschiedliche Ansätze und Abstrahierungsgrade. Alle vorgestellten Spiele zeigen die Verkehrsmittel mit einer Außenperspektive. Drei der vier vorgestellten Spiele basieren auf einer Gott-Ansicht, die eine gute Übersicht über die Spielszenen erlaubt. BusMeister basiert auf einer Parallelperspektive, die eine Straße von der Seite zeigt und gut für eine lokal begrenzte Simulation auf Mikroebene geeignet ist.

Der Abstrahierungsgrad ist bei Mini Metro am größten. Das Spiel beruht auf einfachen geometrischen Formen. Ein Nachteil von dieser hohen Abstraktion ist, dass zum Beispiel das Verkehrsmittel nicht mehr erkannt wird und mit einem Text beschrieben werden muss. Zudem sind die Verkehrssysteme in Mini Metro nur anhand der Levelbeschreibungen ersichtlich. Der Abstrahierungsgrad von B.u.S. muss folglich geringer sein im Vergleich zu Mini Metro.

Alle Spiele haben gemeinsam, dass sie eine direkte oder indirekte Darstellung der Zufriedenheit der Fahrgäste verwenden. Mini Metro und ProtoWorld zeigen die Zufriedenheit indirekt über die Anzeige der Wartezeit der Fahrgäste an. Cities in Motion und BusMeister zeigen die Zufriedenheit direkt über eine Balkenanzeige und Smileys an. Die Darstellungen der Zufriedenheit der Fahrgäste in den Spielen geben den Spielern eine Rückmeldung über den Erfolg der gewählten Spielstrategie.



2 Vorgehensmodell für die Softwareentwicklung

Für eine erfolgreiche Softwareentwicklung ist ein gutes Vorgehensmodell notwendig. Das Vorgehensmodell beschreibt den Ablauf der Softwareentwicklung. Für die Entwicklung von Serious Games sind Ressourcen, Zeit und Budget häufig begrenzt (Pagulayan u. a., 2007). Demnach ist es von zentraler Bedeutung, das Spiel möglichst schnell und gut geplant zu entwickeln. Andererseits müssen Spieltester möglichst früh in den Entwicklungsprozess eingebunden werden. Daher wurde für die Entwicklung von B.u.S. ein auf diese Rahmenbedingungen angepasstes Vorgehensmodell verwendet. Die nachfolgenden Kapitel analysieren bestehende Vorgehensmodelle und beschreiben die iterative Entwicklung.

2.1 Grundlagen Vorgehensmodelle

Ein Vorgehensmodell ist nach Brandt-Pook & Kollmeier (2008) eine Darstellung eines Ablaufs mit den folgenden Merkmalen:

1. Festlegung des Ablaufs von Projekten gleicher Art
2. Benennung von Beteiligten und Aufgaben
3. Methoden, die zur Bewältigung der Aufgaben benutzt werden

Vorgehensmodelle dokumentieren Abläufe durch die Definition von Phasen. Eine Phase fasst dabei eine gewisse Anzahl von Tätigkeiten zusammen. Die Modelle beschreiben das Vorgehen in mehreren gleichartigen Projekten. Das heißt, sie enthalten keinen definierten Zeitplan, sondern nur verallgemeinerte Pläne.

Die Motivation für Vorgehensmodelle ist die Zerlegung des Ablaufs in Teilschritte mit kleineren und beherrschbaren Aufgaben. Sie vereinfachen die Kommunikation durch Einführung

von standardisierten Phasen. Sie können ebenso als Qualitätsmerkmal dienen, um Vergleichbarkeit zwischen Projekten zu erreichen (Brandt-Pook & Kollmeier, 2008).

Die Definierung der Beteiligten geschieht, wie auch der Zeitplan, nur allgemein. Die direkte Benennung von Personen ist nicht Bestandteil eines Vorgehensmodells. Stattdessen soll es Rollen und Aufgabenbeschreibungen für die Beteiligten definieren.

Das dritte Merkmal beschreibt, dass Auflistungen von Methoden und Anweisungen zum Erreichen der Ziele enthalten sein sollen. Jeder Phase kann Methoden mitgeliefert, die sich bei Umsetzungen bewährt haben.

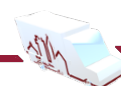
Das Basismodell nach Brandt-Pook & Kollmeier (2008) sieht drei große Bereiche für ein Vorgehensmodell vor:

- Projektmanagement-Bereich
- Kernbereich mit den Phasen Auftragsklärung, Konzeption, Design, Realisierung, Testen und Einführung
- Projektinfrastruktur

Die ersten Vorgehensmodelle in der Softwareentwicklung waren sequentielle Modelle mit einem linearen Ablauf. Diese sind nur auf die Realisierung einer Software ausgelegt und haben Analysen, Evaluationen und Verbesserungen nicht favorisiert. In sequentiellen Modellen sind Schleifen bei der Softwareentwicklung nur zu der vorherigen Phase oder innerhalb einer Phase vorgesehen. Iterative Modelle legen einen Schwerpunkt auf dynamische Reaktionen durch die Nutzung von Analysen und Evaluationen. Das Feedback der Nutzer können Entwickler bei der Nutzung eines iterativen Modells innerhalb kurzer Zeit in das Projekt einarbeiten. Nachfolgend werden verschiedene populäre iterative Vorgehensmodelle erläutert.

2.2 Iterative Entwicklung

Iterative Vorgehensmodelle teilen die Entwicklung in mehrere Iterationsphasen auf. Der Entwicklungsumfang kann aufgeteilt und verteilt werden (Kleuker, 2013). Die Iterationen werden nacheinander durchgeführt, das Ziel jeder Iteration ist eine bessere stabile funktionsfähige Version des Systems zu entwickeln. Jede Iteration hat das Ziel Feedback zu erhalten, welche in das System eingearbeitet werden kann. Eine Änderung bei den Anforderungen kann bei einer iterativen Entwicklung einfacher eingearbeitet werden als bei einer sequentiellen.



Das wohl einfachste iterative Vorgehensmodelle ist in Abbildung 2.1 zu sehen. Dieses Modell beschreibt den essentiellen Entwicklungsprozess. Es besteht aus fünf Phasen (Anforderungsanalyse, Grobdesign, Feindesign, Implementierung, Test und Integration) und einer Iterationsschleife. Damit bildet es die Kernbestandteile der komplexeren Vorgehensmodelle ab. Der Pfeil von der Test- und Integrationsphase zur Phase der Anforderungsanalyse ist der Teil der Iteration, der die dynamische Reaktion und die Anpassung der Anforderungen darstellt. (Kleuker, 2013)

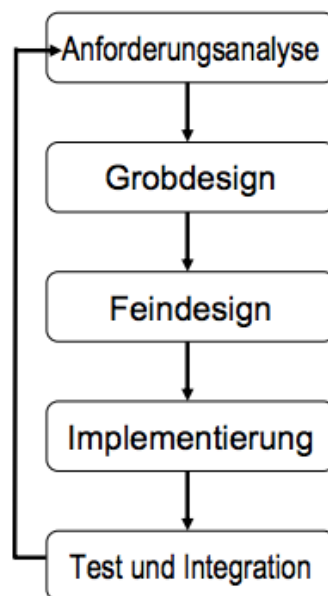


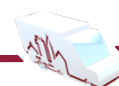
Abbildung 2.1: Iteratives Vorgehensmodell mit fünf Phasen (Kleuker, 2013, S. 31)

2.2.1 Spiralmodell nach Boehm (1988)

Das Spiralmodell ist ein Vorgehensmodell mit einer spiralförmigen Anordnung der Teilprozesse der Softwareentwicklung. Es wurde erstmalig 1986 von Barry W. Boehm beschrieben und ist eine Weiterentwicklung des sequentiellen Wasserfallmodells (Brandt-Pook & Kollmeier, 2008). Im Gegensatz zu sequentiellen Entwicklungsmodellen arbeitet das Modell mit Iterationen mit zyklischen Wiederholungen von Teilprozessen. Die Zyklen sind gleichartig aufgebaut.

Wie in Abbildung 2.2 zu sehen ist, ordnet das Modell jedem Teilprozess einen der folgenden vier Quadranten zu:

1. Festlegen der Ziele und Rahmenbedingungen



2. Beurteilen von Alternativen und Risikoanalyse der Prototypen
3. Entwicklung und Testen der Software, Umsetzung der besten Alternative
4. Planung des nächsten Entwicklungszyklus

Das Modell hat einen Schwerpunkt bei der Risikobetrachtung. In jedem Zyklus werden die jeweiligen Risiken betrachtet und abgeschätzt. Da die Risiken bei neuen Technologien nur schwer abschätzbar sind, besteht bei Nutzung des Spiralmodells die Möglichkeit einer Kostenexplosion (Boehm, 1988).

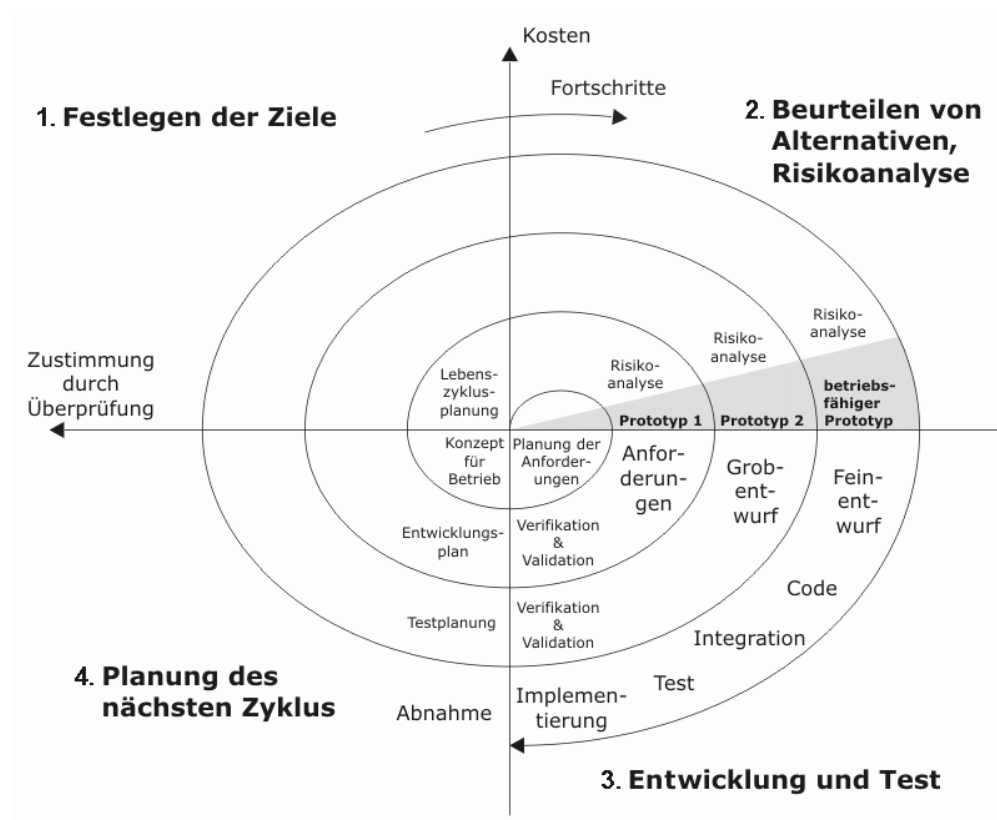
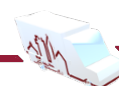


Abbildung 2.2: Spiralmodell nach Boehm (Boehm, 1988)

2.2.2 Playtesting-Modell

Das Playtesting-Modell nach Fullerton (2008), welches in Abbildung 2.3 zu sehen ist, ist ausgelegt auf wiederholende Evaluierung und Testen eines Prototyps. Dieses Vorgehensmodell ist für die Entwicklung von Spielen ausgelegt. Fullerton (2008) empfiehlt eine möglichst



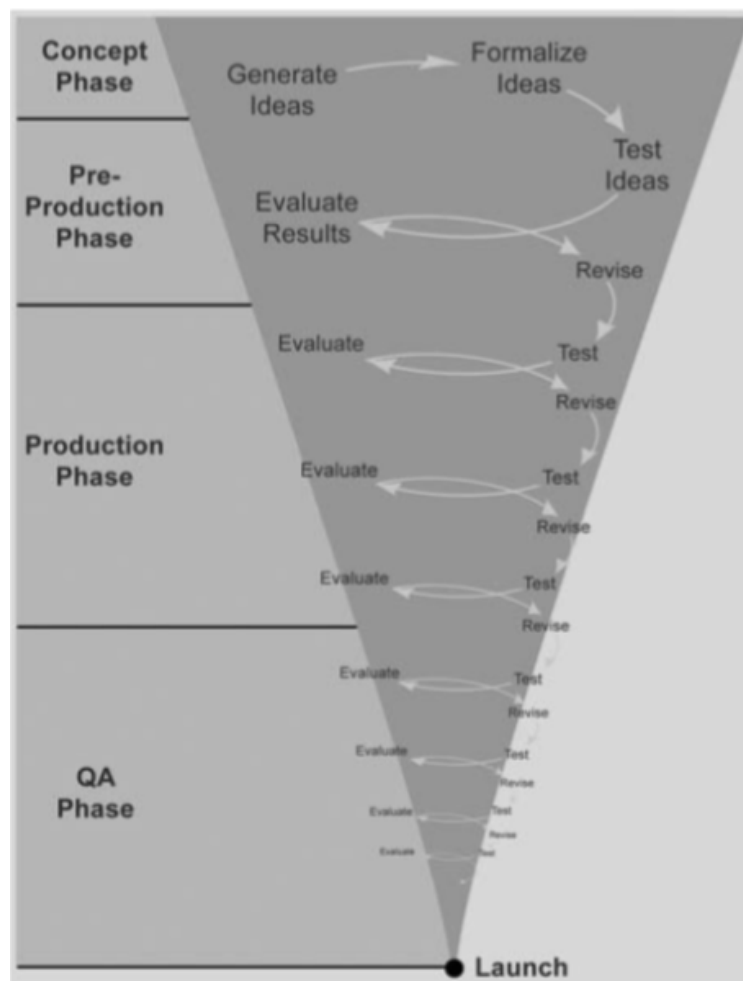
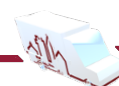


Abbildung 2.3: Playtesting-Modell (Fullerton, 2008, S. 249)

frühe prototypische Umsetzung und anschließende Evaluation des Spielkonzepts. Das Modell beinhaltet dies und zielt auf eine kontinuierliche Evaluation des Spielkonzepts ab. Die Evaluationen sorgen für eine zielgruppenorientierte Entwicklung. In jeder Iterationsphase wird mindestens eine Softwareversion entwickelt. Die erste Phase sieht die Entwicklung einer Idee vor. Ab der zweiten Phase sind Evaluierungen in jeder Phase eingeplant. Die Testpersonen nennt Fullerton (2008) Playtester. Dies sind Personen, die das Spiel testen und zudem Feedback über das Spielerlebnis geben. Des Weiteren empfiehlt Fullerton (2008) eine Beobachtung der Tester, um die Spielsituationen zu identifizieren, bei denen die Tester Probleme haben, bei denen Frustration oder Langeweile aufkommt.

Mit Fortschritt des Projekts sollten die Iteration kleiner werden und die Design-Änderungen geringer werden. Größer werdende Änderungen in Verlauf des Projekts können ein Scheitern des Projekts andeuten.



2.2.3 Game Development Life Cycles

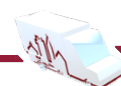
Neben den Standardvorgehensmodellen für die Softwareentwicklung gibt es angepasste Vorgehensmodelle speziell für die Spieleentwicklung, sogenannte Game Development Life Cycles (GDLCs) (Ramadan & Widyani, 2013). Es gibt verschiedene GDLCs, die häufig von Spielstudios entwickelt und benutzt werden. GDLCs basieren auf dem sogenannten Software Development Life Cycle, der nach Ramadan & Widyani (2013) typischerweise die vier Phasen Analyse, Design, Code und Testen enthält.

Ramadan & Widyani (2013) zählen zum Beispiel vier Referenz-GDLCs, von denen zwei GDLCs Iterationen enthalten. Der Doppler Interactive GDLC enthält sechs Phasen: Design, Develop, Evaluate, Test, Review und Release. Er enthält eine iterative Schleife zwischen Develop und Evaluate und eine iterative Oberschleife zwischen Review und Design.

Der Heather Chandler's GDLC sieht hingegen nur vier Phasen vor: Pre-Production, Production, Testing und Post-Production. Alle Phasen sind innerhalb einer Schleife angeordnet.

Aufbauend auf den Referenz-GDLCs haben Ramadan & Widyani (2013) einen neuen GDLC entwickelt, welcher in Abbildung 2.4 zu sehen ist. Dieser beinhaltet die sechs Phasen Initiierung, Pre-Production, Production, Testing, Beta und Release. Die drei Phasen Pre-Production, Production und Testing sind in einer Iterationsschleife angeordnet. Diese Iterationsschleife wird hier Produktionszyklus genannt.

Aleem u. a. (2016) haben die Anzahl und die Inhalte von verschiedenen GDLCs untersucht. Dabei wurden allein in dieser Studie 148 verschiedene Paper mit verschiedenen GDLCs untersucht. Diese Studie zeigt die Vielfalt an Vorgehensmodellen mit verschiedenen Ansätzen.



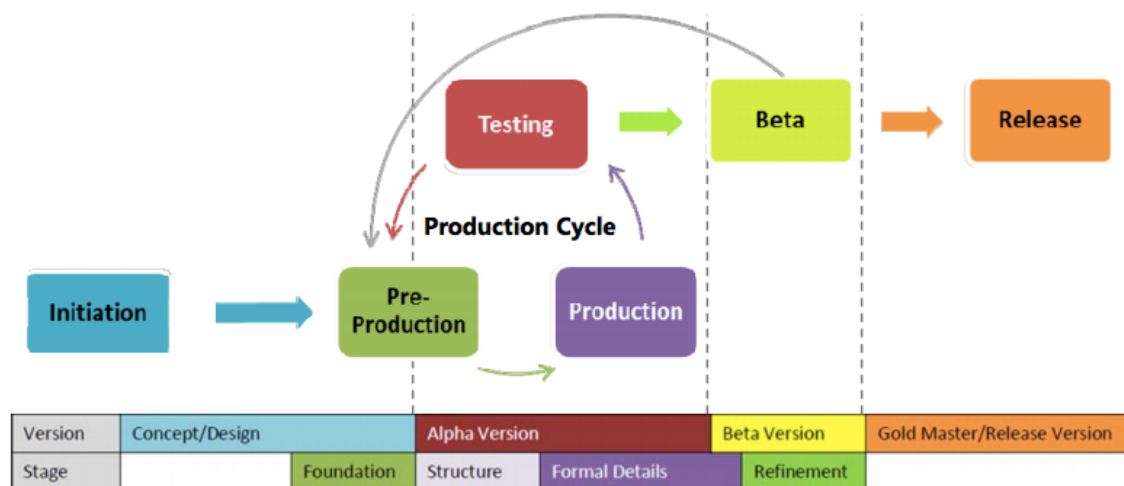
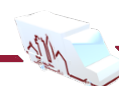


Abbildung 2.4: Beispiel eines Game Development Life Cycles (Ramadan & Widyani, 2013)

2.2.4 Vorgehensmodell für Serious Games

Das nachfolgend vorgestellte Vorgehensmodell ist speziell für die Entwicklung von Serious Games vorgesehen (Strahringer & Leyh, 2017). Die Abbildung 2.5 zeigt das Modell mit seinen sechs Phasen, die sequentiell angeordnet sind. Das Modell beinhaltet die Phasen Exploration, Analyse, Idee, Design, Produktion und Go Live. In der ersten Phase legen die Stakeholder die Funktionen und Ziele des Projekts fest. Die nächsten drei Phasen dienen der Ideenfindung und Ausgestaltung. Anschließend beschreiben die letzten beiden Phasen ein Produktionsprozess und die Einführung der Software. Alle Phasen nutzen einen Methodenkatalog, der auf dem Anwendungsfall angepasst sein sollte. Jede Phasen baut auf einem Input–Throughput–Output-Schema auf. Es sind Iterationen vorgesehen in den einzelnen Phasen, allerdings sind keine Iterationen über mehrere Phasen vorgesehen. Das Modell hat einen Schwerpunkt bei der Design- und Konzeptfindung. Vor der Produktionsphase gibt es vier Phasen, die diesen Schwerpunkt behandeln. Für den Produktionsprozess mit der Softwareentwicklung und dem Softwaretest gibt es nur eine einzige Phase. (Strahringer & Leyh, 2017)

Für jede Phase sind Ein-, Ausgangsdokumente und spezifische Rollen vorgesehen, welche in Abbildung 2.5 nicht enthalten sind. Die Dokument- und Rollenstruktur ist für größere Entwicklungsteams gedacht und kann auf kleinere Projekte schlechter angewendet werden.



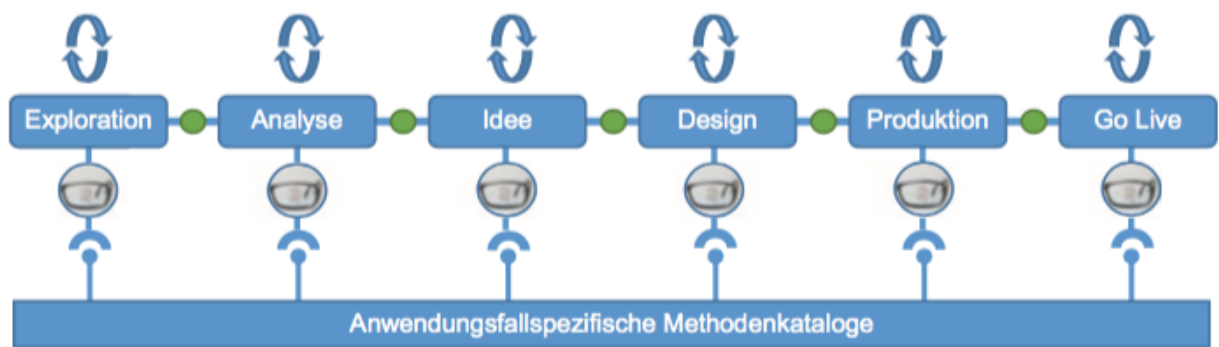


Abbildung 2.5: Vorgehensmodell für Serious Games (Strahringer & Leyh, 2017, S. 20)

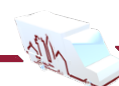
2.3 Auswahl eines Vorgehensmodells

Dieses Kapitel beschreibt die Anforderungen für ein Vorgehensmodell und die Analyse der vorgestellten Vorgehensmodelle für die Entwicklung von B.u.S.

2.3.1 Anforderungen an ein Vorgehensmodell für das Projekt B.u.S.

Die Aufgabenstellung sieht eine iterative Entwicklung vor. Das Feedback von Testern soll direkt in die Entwicklung mit einfließen. Bei der iterativen Entwicklung soll es mehrere Evaluationen, mehrere Auswertungsphasen und mehrere Implementationsphasen geben. Das Vorgehensmodell muss den kompletten Entwicklungsprozess abbilden. Die Phasen in dem Modell müssen an die Gegebenheiten anpassbar sein. Wichtig ist bei dem Serious Game so beispielsweise die Erstellung und Beachtung eines didaktischen Konzepts.

Für die Entwicklung von B.u.S. gibt es eine begrenzte Zeitspanne und nur einen verfügbaren Softwareentwickler. Diese Bedingungen muss ein potentiell Vorgehensmodell ebenfalls beachten. Daraus ergibt sich, dass es keine komplizierte Rollen- und Dokumentenstruktur besitzen darf. Diese würde für einen einzelnen Entwickler ein zu großes Overhead darstellen. Der Schwerpunkt des Vorgehensmodells muss auf der Produktionsphase, auf den Evaluationen und auf der Verarbeitung des Feedbacks liegen.



2.3.2 Analyse der vorgestellten Vorgehensmodelle

Das erste vorgestellte iterative Vorgehensmodell ist das Spiralmodell nach Boehm. Dieses Modell hat einen Schwerpunkt bei der Risikobetrachtung. Für die Entwicklung von B.u.S. ist allerdings ein Schwerpunkt bei der Produktions- und Evaluationsphase vorgesehen.

Der Game Development Life Cycle ist speziell für die Spieleentwicklung ausgelegt. Das Modell nach (Ramadan & Widyani, 2013) sieht auch eine iterative Entwicklung vor, muss allerdings auf die Gegebenheiten angepasst werden.

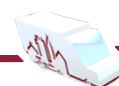
Das Vorgehensmodell nach Strahringer & Leyh (2017) für Serious Games ist speziell für die industrielle Entwicklung von Serious Games gedacht. Dieses sieht ein größeres Team und verschiedene Rollen in dem Team vor. Es ist weniger gut für kleinere Projekte geeignet.

Das Playtesting-Modell nach Fullerton (2008) hat einen Schwerpunkt auf der iterativen Entwicklung und sieht mehrere Evaluationen vor. Damit ist es gut geeignet für die Entwicklung von B.u.S. Die Phasen und die Inhalte der Iterationen müssen allerdings noch angepasst werden.

2.3.3 Angepasstes iteratives Vorgehensmodell zur Entwicklung und Evaluation von Serious Games

Das Vorgehensmodell zur Entwicklung von B.u.S. basiert auf dem Vorgehensmodell nach Fullerton (2008) und enthält insgesamt vier Phasen. Die vier Phasen teilen sich auf in Konzeptphase, Analyse, Evaluationen, Qualitätsmanagement und werden nachfolgend erläutert. Der Output der jeweiligen Phasen ist der Input für die jeweilige nächste Phase.

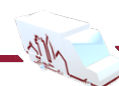
- **Konzeptphase:** Die erste Phase ist die Konzeptphase, in der die Exploration des Projekts und die Aufstellung eines didaktischen Konzepts erfolgt. Der Output in dieser Phase ist neben dem didaktischen Konzept, ein erstes Design für das Serious Game.
- **Analysephase:** Nach der Konzeptphase folgt die Analyse-Phase mit einem Fokusgruppen-Gespräch für die weitere Konzeptfindung. Hier werden neue Ideen generiert und das erste Konzept wird überarbeitet. Mehrere Iterationen sind für B.u.S. in der Phase nicht vorgesehen, sind aber theoretisch möglich. Der Output ist ein überarbeitetes Konzept und neue Anforderungen, die anschließend implementiert werden können.



- **Evaluationsphase:** Nach der Analysephase folgt die Evaluationsphase in der das Konzept überprüft und iterativ verbessert wird. Wie auch die Analysephase sind die Iterationen in dieser Phase dreigeteilt in Implementierung, Durchführung der Evaluation und Auswertung der Evaluation. Die Evaluationsphase enthält Usability-Tests in Form von Thinking Aloud Tests und heuristische Evaluationen. Diese beiden Evaluationsmethoden haben ähnliche Ziele und können in beliebiger Reihenfolge durchgeführt werden. Das Ziel dieser Arbeitsweise ist ein hoher Reifegrad des Prototypen und damit auch des Konzepts. Dieses Konzept kann anschließend in die finale Software überführt werden.
- **Qualitätsmanagementphase:** Abschließend folgt die Qualitätsmanagement-Phase, welche die notwendigen Software- und Abnahmetests beinhaltet. Die Softwaretests sollen die innere Qualität der Software sicherstellen.

Neben den vier Phasen gibt es das Projektcontrolling, das parallel zu den Phasen über das ganze Projekt hinweg aktiv ist. Zu dem Projektcontrolling gehören in diesem Fall das Anforderungs- und Zeitmanagement.

Die Abbildung 2.6 zeigt das angepasstes Entwicklungsmodell. Es besteht aus den vier genannten Phasen und hat vier projektspezifische Iterationen.



Iteratives Entwicklungsmodell

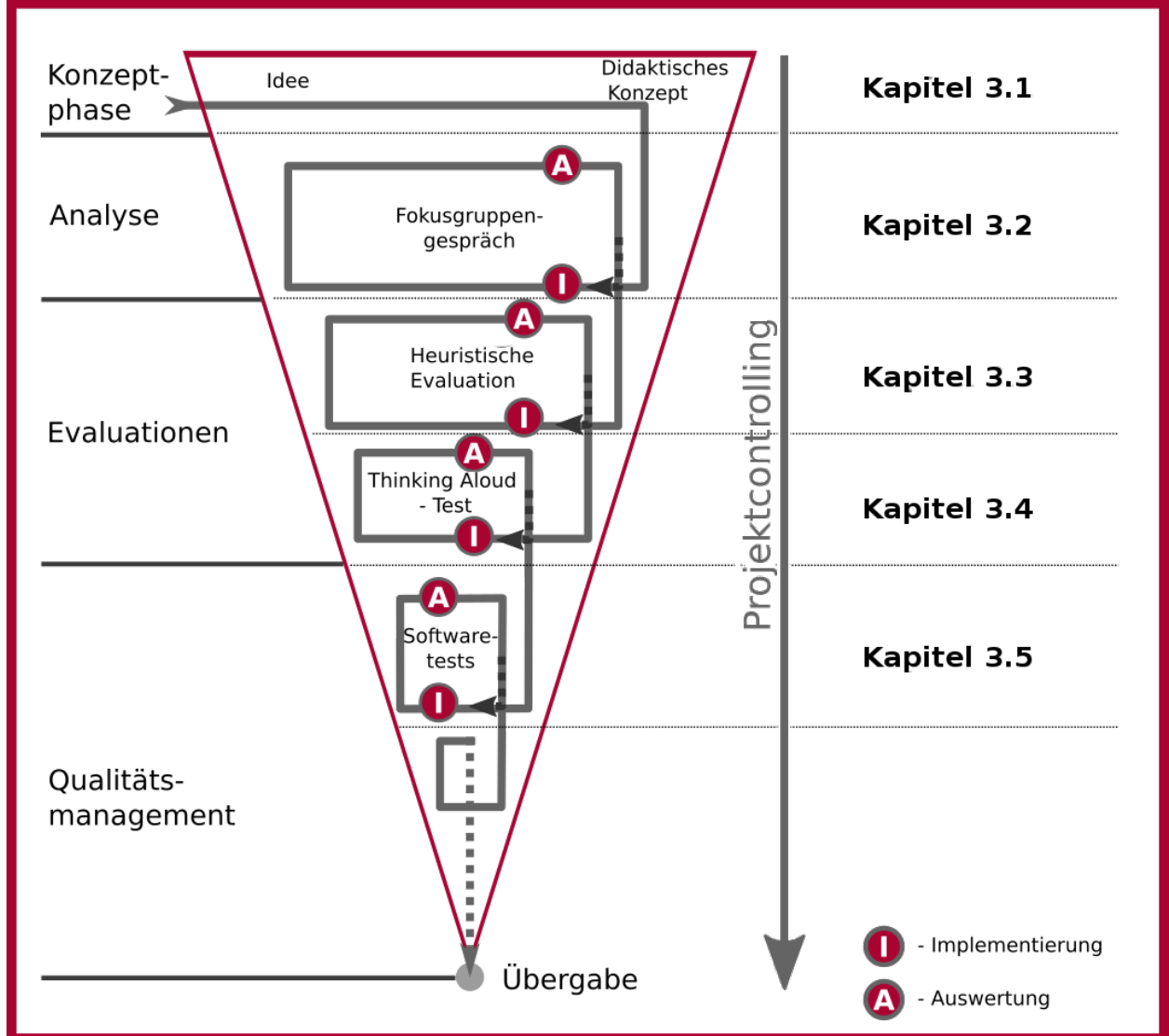
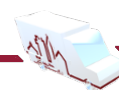


Abbildung 2.6: Angepasstes iteratives Entwicklungsmodell (eigene Abbildung)



3 Iterative Entwicklung

3.1 Iteration 1 - Konzeptphase

Das Serious Game wird mit dem Ziel entwickelt, das Konzept eines Mobility-On-Demand-Busses zu vermitteln. Deshalb wurden zu Beginn der Entwicklung die didaktischen Anforderungen festgelegt. Die nachfolgenden Unterkapitel erläutern das didaktische Konzept und die abgeleiteten Anforderungen.

3.1.1 Methoden

Didaktisches Konzept

Das didaktische Konzept hat die Aufgabe Lerninhalte zu definieren und zu strukturieren. Nach Zyda (2005) wird der didaktische Ansatz benötigt, um die Lernziele schon während der Entwicklung des Serious Games zu berücksichtigen.

Die klassische Didaktik läuft nach Terhart (1997) in vier Schritten ab:

- Festlegen des Soll-Zustands
- Ermittlung und Analyse des Ist-Zustands
- Planung des Übergangs von Ist- nach Soll-Zustand
- Umsetzung der Planung in Form des Unterrichts

Diese klassische Didaktik kann bei einem Serious Game mit einer großen Zielgruppe nicht umgesetzt werden, da der Ist-Zustand des Wissens nicht ermittelt werden kann. Die Ermittlung des Ist-Zustands müsste im Spiel und automatisiert erfolgen. Dies ist nahezu unmöglich, da ein ausführlicher Fragebogen im Spiel oder andere ähnliche Methoden das Spielerlebnis

zerstören würden. Bedingt durch die Größe der möglichen Zielgruppe für B.u.S., ist der Ist-Zustand sehr uneinheitlich.

Nach Prensky (2007) sind Serious Games dennoch in der Lage die Effektivität des Lernens zu steigern, da sie die Nutzer motivieren eigenständig zu lernen und Wiederholungen anregen.

Die Lehrmethode, die für B.u.S. genutzt werden soll, ist das explorative Lernen. Bei dem explorativen (auch entdeckendes Lernen genannt) wird davon ausgegangen, dass der Lernende die Inhalte selbst entdeckt und seinen Lernprozess selbst steuert (Klix & Spada, 1998). Das heißt, es gibt keine externen Lehrer, die Aufgabenlösungen erarbeiten die Nutzer selbst ohne äußere Hilfe. Dabei soll das Serious Game allerdings das Thema, Lerninhalte und Lernelemente vorgeben. Das Serious Game leitet die Nutzer durch die Vorgabe von Aufgaben. Die Nutzer bekommen dabei den Freiraum, eigene Hypothesen für die Aufgaben zu testen. Falsche Hypothesen bei den Aufgaben führen nicht zu einem Erfolg, ein virtueller Assistent soll den Nutzer bei der Reflektion der eigenen Hypothesen unterstützen.

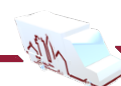
Das explorative Lernen mit einer mobilen App sorgt für flexibles, lernortunabhängiges Lernen, dass jederzeit möglich sein, aber dennoch inhaltlich durch die App geleitet wird. Nebenbei steigert es den Spielspaß der Nutzer durch eine intrinsische Motivation (Klix & Spada, 1998).

Die intrinsische Motivation ist die Motivation, die aus eigenen Antrieb erfolgt (Klix & Spada, 1998). Das Gegenteil der intrinsischen Motivation ist die extrinsische Motivation. Diese wird aufgezwungen durch einen externen Einfluss, wie zum Beispiel eine Belohnung von außen. Bezogen auf Serious Games bedeutet eine intrinsische Motivation, dass das Spiel spannend und interessant wirkt, um die Neugier der Nutzer anzusprechen. Eine extrinsische Motivation kann erreicht werden durch Belohnungen und freigeschaltete Elemente bei einem Erfolg im Spiel.

Game-Design-Elemente

Game-Design-Elemente sind notwendige Elemente für das Design eines Spiels. Die wichtigsten Game-Design-Elemente sind das Spielziel, die Spielregeln und der Game Loop (Rehfeld, 2013).

Das Spielziel ist das Ziel, das der Spieler erreichen soll. Ein Spielziel kann aus mehreren Gewinnkonditionen bestehen und unterscheidet sich von der Gewinnsituation. Bei einem Serious Game kann das Spielziel der Gewinn von Wissen sein. Die Gewinnkonditionen können individuell sein und können sich mit dem Spielverlauf auch ändern. Komplexe Spiele haben



häufig eine Stufung bei den Spielzielen (kurzfristige, mittelfristige und langfristige Spielziele) und können zum Teil auch endlos weitergehen (Rehfeld, 2013).

Die Spielregeln bestimmen die Relationen zwischen den Spielelementen. Sie sorgen für einen geregelten Spielablauf und haben Einfluss auf den Spielfluss (Rehfeld, 2013).

Bei vielen Spielen wiederholen sich die Aktionen innerhalb einer Spielrunde, die sogenannte Game Loop. Die Abfolge der Aktionen kann in den Spielregeln festgelegt sein. Bei digitalen Spielen hingegen wird die Reihenfolge der Aktionen häufig dem Spieler überlassen. Die Spielrunden sind dann nicht mehr klar abgrenzbar (Rehfeld, 2013).

MDA - Framework

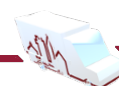
Die MDA-Framework wurde zwischen 2001 und 2004 beim Game Design and Tuning Workshop auf der Game Developers Conference in San Jose entwickelt. Die Buchstaben MDA stehen bei dem Framework für Mechaniken, Dynamiken und Ästhetiken. Hunicke u. a. (2004) haben das Framework als formales Modell für das Spieldesign entwickelt.

Die Abbildung 3.1 zeigt das zu Grunde liegende Konzept des Frameworks. Hunicke u. a. (2004) gehen davon aus, dass der Designer eine andere Sicht auf das Spiel hat, als der Spieler. Der Designer hat eine Produktsicht. Er erstellt das Spiel mit den Spielelementen und den Spielregeln. Der Spieler konsumiert das Spiel und dessen Inhalte. Das Framework beschreibt die Wirkung von Relationen von Spielelementen auf dem Spieler. Der Konsum basiert auf dem Spielsystem, das auf Regeln aufbaut. Aus der Sicht des Spieldesigners basiert das Spiel auf Mechaniken. Die Mechaniken bilden die Datengrundlage und die Logik des Spiels. Die Mechaniken erzeugen Dynamiken, die zur Laufzeit das Spiels steuern und den Spielfluss verändern. Der Spieler empfindet die Ästhetiken des Spiels bei der Interaktion mit dem Spiel.

Ästhetik

Die Ästhetik des Spiels generiert den Spielspaß. Der Spielspaß bei einem Spiel basiert auf Motivationen die Hunicke u. a. (2004) wie folgt differenziert haben:

- *Spielsensation*: Wohlfühlen im Spiel
- *Fantasie*: Eintauchen in Traumwelten
- *Narrative*: Geschichte oder Erzählung



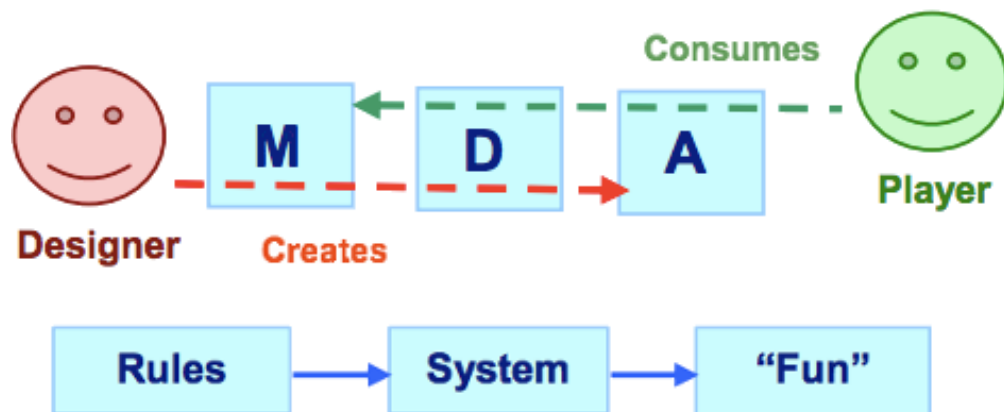


Abbildung 3.1: MDA-Framework nach Hunicke u. a. (2004)

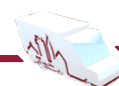
- *Herausforderung*: Spieler muss sich beweisen
- *Fellowship*: Positives Erlebnis bei Gemeinschaft
- *Discovery* oder *Explorative*: Entdecken und Erforschen der Spielwelt
- *Expression*: Selbstwirksamkeitserfahrung
- *Submission*: Unterwerfung und Ablenkung durch das Spiel

Dynamiken

Die Mechaniken erzeugen Dynamiken, die zu einem Spielfluss führen. Dieser soll dabei unterstützen, dass der Spieler weiter spielen will.

Mechaniken

Die programmierte Logik des Spiels bildet die Mechaniken ab. Die Logik basiert auf den Spielregeln. Die Spielregeln und damit auch die Mechaniken bilden Hierarchien und Verhältnisse zwischen den Spielobjekten ab. Die Logik kann zum Beispiel als State Machine umgesetzt werden. Die Eingaben wirken auf die State Machine, die State Machine berechnet daraufhin auf Basis des aktuellen States den nächsten State und erzeugt die Ausgaben. (Rehfeld, 2013)



3.1.2 Didaktische Anforderungen

Die didaktischen Anforderungen können in drei Bereiche unterteilt werden. Als erstes soll B.u.S. die Funktionsweise des Rufbusses zeigen. Es soll gezeigt werden, dass der Rufbus ein zeitlich und räumlich flexibler Zubringer ist, der nur bei einer gewünschten Nutzung kommt. Der Rufbus hat durch die virtuellen Haltestellen eine höhere Haltestellendichte.

Des Weiteren soll das Spiel die Vorteile von Rufbussen erläutern. Diese sind eine hohe Beförderungsleistung, kurze Zuwege dank der erhöhten Haltestellendichte und kurze Wartezeiten, da der Bus vorher gebucht wird. Für den ÖPNV-Planer hat der Rufbus noch einen weiteren Vorteil: Der Bus absolviert weniger Leerfahrten und ist damit umweltfreundlicher.

Gleichzeitig soll dem Spieler bewusst werden, dass Rufbusse nur sinnvoll eingesetzt werden können bei Stationen mit einem geringen Beförderungsbedarf. In dem Serious Game müssen neben dem Rufbus die Vorteile des Linienbusverkehrs gegenüber dem Rufbus auch erwähnt werden. Der Rufbus kann den Linienbusverkehr nicht vollständig ersetzen, da dieser für höhere Beförderungsleistungen benötigt wird.

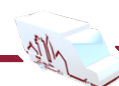
Nicht notwendig für B.u.S. ist die Darstellung von anderen Beförderungsmöglichkeiten und anderen Verkehrsteilnehmer wie zum Beispiel Fahrradfahrer, Bahnen oder Autos. Eine Verkehrssteuerung zu beachten ist ebenfalls nicht vorgesehen.

3.1.3 Spielmodell

Für die Entwicklung eines Spielmodells wurde das MDA-Framework benutzt, welches in Kapitel 3.1.1 erklärt wurde. Mit Hilfe des Frameworks kann das Spiel mit einem Top-Down-Ansatz zerlegt werden. Das heißt, dass nachfolgend das Spielmodell von der Spielersicht aus beschrieben wird.

Ästhetik

Von den acht genannten Ästhetiken soll B.u.S. vor allem die Ästhetiken *Challenge* und *Discovery* ansprechen. Der Spieler soll Herausforderungen durch Aufgaben in einzelnen Levels bekommen. Für eine gleichbleibende Herausforderung ist eine gleichmäßige Steigerung des Schwierigkeitsgrad über die Level vorgesehen. Mit Erhöhung der Level bekommt der Spieler neue Spielelemente und neue Spielziele. Diese Elemente kann der Spieler entdecken, so dass dadurch die Neugier der Spieler befriedigt wird.



Dynamiken

Die Spieldynamiken nutzen die zur Verfügung stehenden Spielelemente und bestimmen die Regeln. Die wichtigsten Spielelemente sind der Rufbus, die Zufallspassagiere, die Bushaltestellen und das Straßennetz.

Der Spieler hat die Aufgabe alle auftauchenden Passagiere zu befördern. Dazu hat er einen Rufbus zur Verfügung. Der Spieler ist in der Rolle des lokalen ÖPNV-Planers, der die Route des Rufbusses plant.

Ein Linienbus ist ebenfalls eingeplant. Dieser soll nicht von dem Spieler gesteuert werden, sondern er soll automatisch einer festgelegten Route folgen.

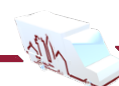
Die Erzeugung der Ästhetik *Challenge* wird erreicht, in dem der Spieler Ziele erfüllen muss. Für die Durchführung hat dieser einen begrenzten Zeitrahmen.

Mechaniken

Das Spiel soll Passagiere automatisch generieren und verteilen. Falls ein Rufbus oder ein Linienbus an einer Busstation ist, sollen diese Passagiere in den Bus einsteigen. Sobald der Bus mit den Passagieren an einer Endhaltestelle ankommt, sollen die passenden Passagiere aus dem Bus aussteigen. Zur Überprüfung der Gewinnkonditionen muss das Spiel die Zeit und die Passagiere im Bus messen.

3.2 Iteration 2 - Fokusgruppengespräch

Die zweite Phase des Vorgehensmodells sieht eine Fokusgruppe mit Experten auf dem Gebiet der Verkehrssystemtechnik und der Softwareentwicklung vor. Diese Phase dient der Konzeptentwicklung. Nachfolgend wird zuerst die Methode mit den Grundlagen erläutert. Anschließend werden das Ziel der Fokusgruppe, die Teilnehmer, die Durchführung und die Auswertung vorgestellt.



3.2.1 Methode

Eine Fokusgruppe, in der Literatur auch Gruppeninterview, Gruppendiskussion oder Workshop genannt, ist eine sozialwissenschaftliche Methode. Nachfolgend werden diese Begriffe gleichbedeutend verwendet. Der Inhalt der Methode ist eine moderne Diskussion in einer Gruppe mit ausgesuchten Teilnehmern zu einem vorgegeben Thema. Das Thema wird durch eine Informationseingabe wie z.B. eine Präsentation oder ein Vortrag eingeleitet. Dabei muss mindestens ein Moderator die Diskussion leiten und den Verlauf mit gezielten Fragen leiten. Die Ergebnisse der Methode sind die Ergebnisse der Interaktionen unter den Teilnehmern. (Schulz u. a., 2012)

Die Stärke von Fokusgruppen liegt in der Exploration eines Themas und in dem Generieren von Daten. Als weiterer Vorteil kann die Flexibilität genannt werden. Fokusgruppen können zu beliebigen Themen eingesetzt werden, wobei sie ermittelnd oder vermittelnd verwendet werden können. Die Teilnehmer können einander bekannt, unbekannt oder gemischte Personengruppen sein. Fokusgruppen sind weniger geeignet für das Testen von Hypothesen, da die Teilnehmer sich bei der Durchführung gegenseitig beeinflussen. (Schulz u. a., 2012)

Der Aufbau einer Fokusgruppen-Analyse kann grob in drei Teile gegliedert werden: Vorbereitung des Themas und Auswahl der Teilnehmer, Datengewinnung durch die Fokusgruppe, Auswertung der generierten Daten (Salzburg Research Forschungsgesellschaft, o.D.). Diese Gliederung ist auch in Abbildung 3.2 dargestellt.

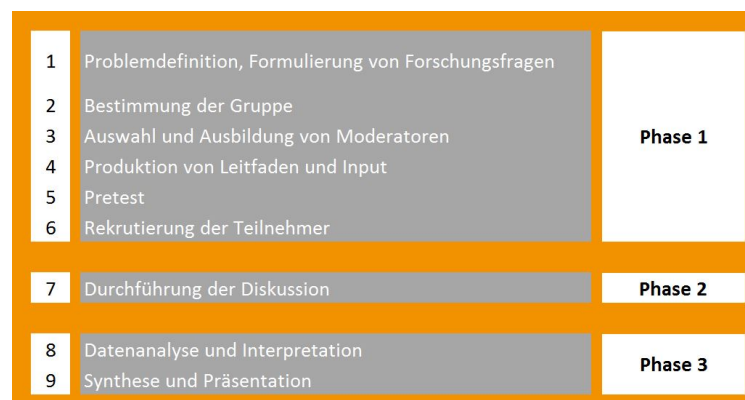
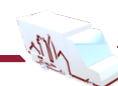


Abbildung 3.2: Ablaufschema eines Fokusgruppenprojekts (Salzburg Research Forschungsgesellschaft, o.D.)



3.2.2 Ziel der Fokusgruppe

Die Fokusgruppe in dieser Arbeit dient der Konzept- und Ideenfindung für das Serious Game. Ein erstes Hauptziel ist die Generierung neuer unabhängiger Ideen für ein neues Spielkonzept. Das Ziel liegt die Annahme zu Grunde, dass unabhängige Teilnehmer unterschiedliche Konzepte entwerfen und unterschiedliche Ideen haben.

Ein zweites Hauptziel ist die Überarbeitung eines bestehenden Konzepts, welches im Vorfeld von der Fokusgruppe erstellt wurde. Zu dem Konzept sollen erste Meinungen von unabhängigen Experten eingeholt werden. Mit diesen Meinungen soll das Konzept verbessert und vervollständigt werden.

3.2.3 Teilnehmer

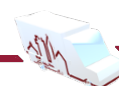
Die Teilnehmer des Fokusgruppengesprächs waren Experten aus dem Bereich Automotive/Verkehr. Es haben drei Studenten und vier wissenschaftliche Mitarbeiter des Bereichs Verkehrssystemtechnik vom DLR an der Fokusgruppe teilgenommen. Von den Mitarbeitern arbeiteten 42% (3 Personen) zwischen zwei und fünf Jahre in dem Bereich, der Rest hatte keine nennenswerte Berufserfahrung. Von den Studenten hat einer Logistik und die beiden anderen Elektronische Systeme in Fahrzeugtechnik, Luft- und Raumfahrt studiert.

Insgesamt befanden sich fünf Männer und zwei Frauen in der Stichprobe. Das Durchschnittsalter betrug zu dem Zeitpunkt der Durchführung 27,1 Jahre mit einer Standardabweichung von 1,7 Jahren. Der jüngste Teilnehmer war 24 Jahre alt, der älteste 29 Jahre.

Alle Personen hatten bereits Erfahrungen bei der Bedienung von Smartphones. Von den Teilnehmern hatten 57% (4 Personen) Erfahrung im Bereich Softwareentwicklung. 71% (5 Personen) spielten täglich Spiele auf dem Smartphone.

3.2.4 Durchführung

Das Fokusgruppengespräch war aufgeteilt in zwei Teile. Der Erste Teil bestand aus der Entwicklung eines eigenen Spielkonzepts in einer Kleingruppe. Zu Beginn des ersten Teils gab es zwei einleitende Diskussionen zu den Themen Spielelemente und Spielspaß. Diese Diskussionen hatten zum Ziel Interaktionen anzuregen und die Teilnehmer auf das Thema einzustimmen.



Der zweite Teil enthielt die Vorstellung und Diskussion eines bestehenden Konzepts für das Serious Game.

Die gesamte Veranstaltung war wie folgt zeitlich gegliedert:

- **Erster Teil:**

- 15 min - Einführung Serious Games und Reallabor Schorndorf
- 15 min - Erklärung von Spielelementen
- 30 min - Teamarbeit in zwei Teams
- 15 min - Vorstellung der erarbeiteten Spielkonzepte

- **Zweiter Teil:**

- 15 min - Vorstellung des eigenen Spielkonzept und der Infotexte
- 20 min - Diskussion des vorgestellten Konzepts
- 10 min - Abschluss der Veranstaltung

Für die Durchführung des ersten Teils des Fokusgruppengesprächs wurden die Teilnehmer in zwei Gruppen aufgeteilt. Die Größe der Gruppe wurde gezielt verkleinert. Durch die kleineren Gruppen sollten ausartende Diskussionen bei der Ideenfindung verringert werden. Ein weiterer Vorteil der beiden Gruppen ist die Entwicklung zweier unabhängiger Spielkonzepte. Da zwei Moderatoren anwesend waren konnten beide Gruppen gleichermaßen betreut werden.

Bei der Durchführung gab es eine Aufnahme mit einem Audiotape, welches durch Notizen ergänzt wurde. Ein Echtzeit-Protokoll wurde nicht mitgeschrieben, da dieses den Redefluss beeinflussen würde und für die Konzeptfindung nicht benötigt wird. Der genaue Wortlaut ist bei den vorgestellten Konzepten nicht relevant.

Das vorgestellte Konzept im zweiten Teil der Veranstaltung ist in Abbildung 3.3 dargestellt. Die Abbildung und die gezeigten Spielelemente wurden bei der Präsentation näher erläutert. Die gezielten Fragen im zweiten Teil des Fokusgruppengesprächs waren in Themengruppen aufgeteilt, wie Krueger (1998) empfohlen hat.

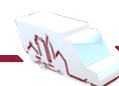




Abbildung 3.3: Gezeigtes Konzept beim Fokusgruppengespräch

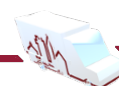
3.2.5 Vorgestellte Konzepte

Die Teams haben bei dem Fokusgruppengespräch erfolgreich zwei verschiedene Konzepte erarbeitet und vorgestellt. Dieses Kapitel stellt die beiden Konzepte in einer Zusammenfassung vor. Das anschließende Kapitel wertet die Konzepte aus, indem es diese vergleicht und die Ideen einordnet.

Team 1 (Schlag den Linienverkehr)

Das erste Team mit dem selbst gegebenen Konzeptnamen „Schlag den Linienverkehr“ hat das Konzept in drei Teile geteilt, den Aufbau des Spiels, die Motivation und die Spielelemente:

- **Aufbau und Menüs des Serious Games:**
 - Menü (Auswahl des Szenarios)
 - Levelbeginn (Liniennetz anzeigen)
 - Randbedingungen der Stadt anzeigen (Ziele des Szenarios, Anzahl Fahrgäste, Anzahl Busse)
 - Planungsphase (Busse und Stationen aus dem Depot einsetzen)
 - Simulation mit Punktesystem (Simulation parallel mit Linienbus, Parameter des Linienbusses sind gegeben)
 - Popup mit Punktesystem nach Ablauf der Zeit



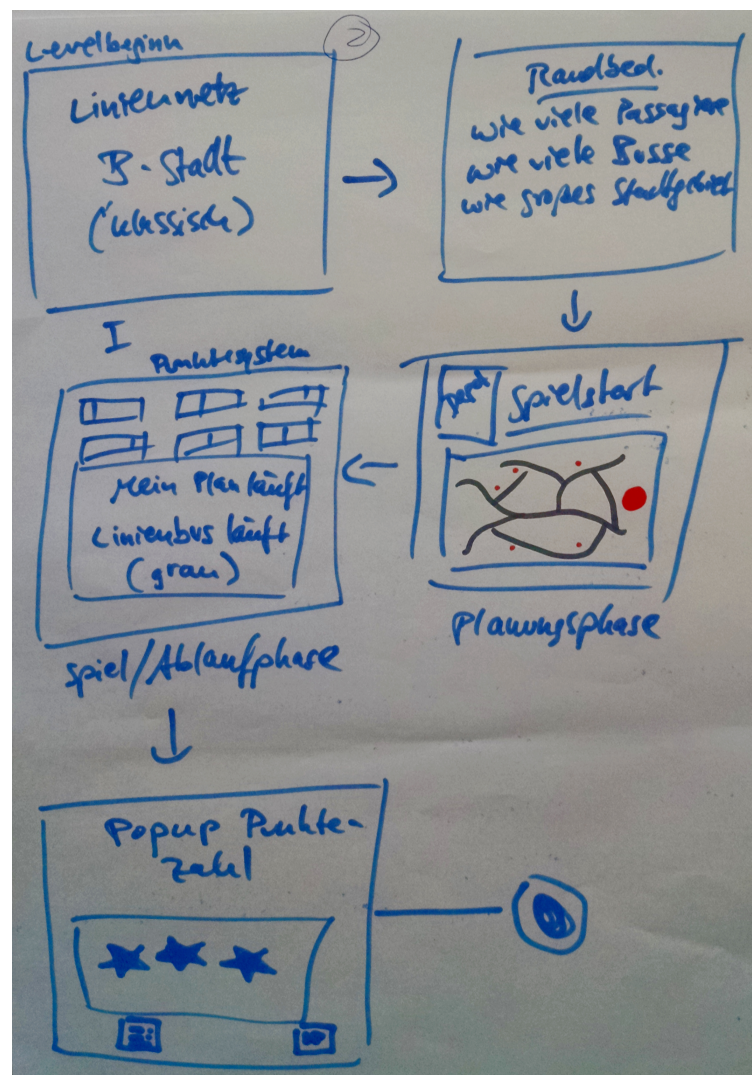


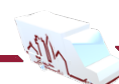
Abbildung 3.4: Konzeptskizze von „Schlag den Linienverkehr“

- **Motivation im Spiel:**

- Besser sein als der Linienverkehr bei verschiedenen Indikatoren
- Verbessern des Busses bei Erfolgen

- **Spielelemente:**

- verschiedene Zeitpunkte (Schwerlastzeit, Schule, Niedriglastzeit)
- Depot mit Rufbus und evtl. Fahrrad und Auto
- alle Faktoren separat abrufbar oder aggregierter Punktestand



- Fahrgäste haben Farben und wollen zu ihrer Farbe hin

Team 2 (Transport Tycoon) Das zweite Team, welches das Konzept „Transport Tycoon“ genannt hat, hat für das Konzept eine Game Loop und die Spielelemente beschrieben. Die Game Loop besteht aus den folgenden Elementen:

- Play (Kaufe, plane, beachte)
- Result (Wartezeit, Einnahmen, Kundenzufriedenheit, Fahrzeit)
- Quest (+Startguthaben)
- Level Up

Eine Konzeptskizze, die während des Fokusgruppengesprächs angefertigt wurde, zeigt Abbildung 3.5.

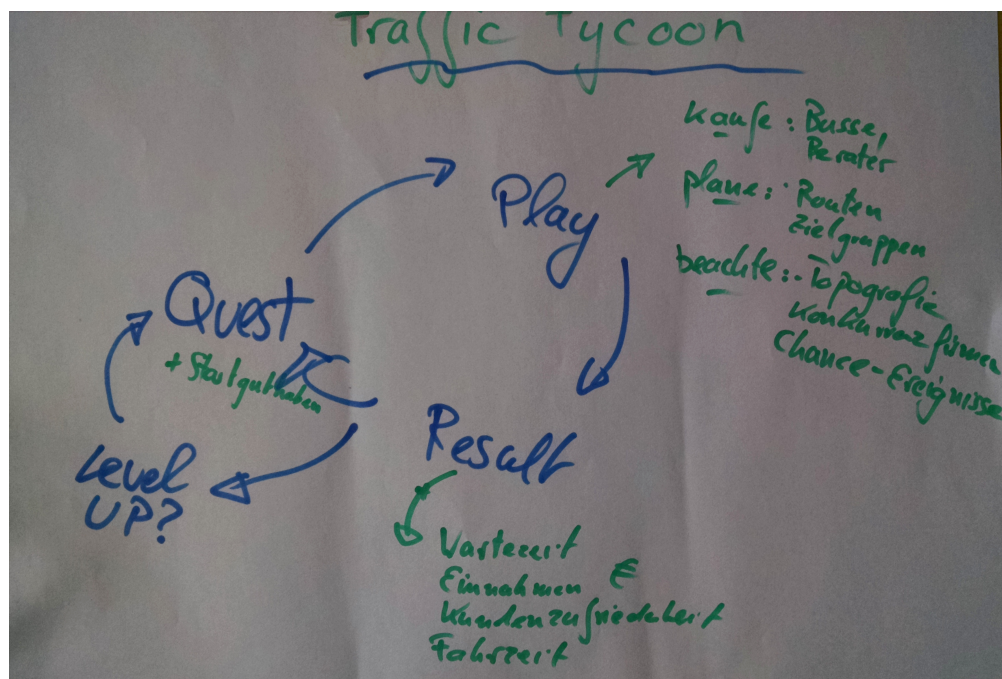
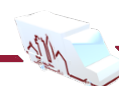


Abbildung 3.5: Konzeptskizze von „Transport Tycoon“

Das Spielkonzept sieht eine Gott-Ansicht auf die Kleinstadt vor, die eine gute Planbarkeit für das levelbasierte Wachstumsspiel gewährleisten soll. Neben der Game Loop hat das Team verschiedene Spielelemente vorgestellt:

- Spielelemente:



- Kleinstadt mit lokalen Busbetreiber und einem Nahverkehrsnetz als Konkurrent
- Rundenziele mit einer begrenzten Zeit (z.B. Beförderung von 50 Fahrgästen in der ersten Woche)
- Shop zum Kaufen von Bussen in verschiedenen Größen
- Kaufen von besseren Beratern (der erster Berater ist kostenlos)
- Routen planen mit den Bussen
- Fahrgäste haben unterschiedliche Charakteristika
- Zufallseignisse: Großereignisse, Sperrungen von Straßen
- durch Beförderungsentgelte und Level kann ein Gewinn erwirtschaftet werden
- Einführung eines Zwangsberaters, falls das Level nicht geschafft wurde
- Bus kostet Geld bei der Fahrt und bei Stillstand
- Einführung eines flexiblen Preissystems erhöht das Fahrgastaufkommen

3.2.6 Auswertung der entwickelten Konzepte

Dieses Kapitel vergleicht die entwickelten Konzepte und zeigt Gemeinsamkeiten und Unterschiede.

Bei der Vorstellung der beiden Konzepte fiel auf, dass die beiden unabhängig voneinander entwickelten Spielkonzepte ein ähnliches Grundkonzept haben. Beide Konzepte basieren auf ein vorgegebenes Straßennetz und werden mit einer Gott-Ansicht bedient. Sie sehen einen Gegenspieler vor und der Spieler muss ein vorgegebenes System schlagen. Dabei spielt der Spieler in einem Level bzw. Spielrunde und ein Punktesystem misst die Performance des Spielers. Als Motivation sehen beide Konzepte eine verdienbare Verbesserung der eigenen Spielelemente vor.

Allerdings gibt es auch Unterschiede bei den Konzepten, wie in der Tabelle 3.1 zu sehen ist.

In einer sehr kurzen Zeit haben beide Teams ein kurzes Spielkonzept entwickeln können. Bedingt durch die kurze Zeit weisen beide Spielkonzepte Lücken auf.

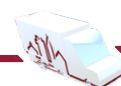


Tabelle 3.1: Gegenüberstellung der Spielkonzepte

Schlag den Linienverkehr	Transport Tycoon
Planungsphase und Simulationsphase	Live-Simulation des Bussystems
Einfacher Berater	Zusätzlicher Berater, Zwangsberater (bei schlechten Spielergebnissen) oder kaufbarer Berater
Punktesystem gibt den Erfolg wieder	Währungssystem mit Gewinnen
Fahrgäste wollen zu einem bestimmten Ziel	Fahrgäste haben Charakteristika
-	Leerstände und Fahrten kosten Geld
Verschiedene Lastzeiten	Störende Großereignisse

3.2.7 Meinungen

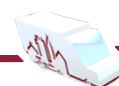
In dem zweiten Teil des Fokusgruppengesprächs wurde ein erstes Spielkonzept für B.u.S. vorgestellt. Zu diesem Konzept gab es sehr viele Anmerkungen. Diese sind nach Themen geordnet nachfolgend zusammengefasst:

Spielkonzept

Das vorgestellte Spielkonzept wurde von allen Teilnehmer als nicht intuitiv angesehen. Vor allem die grafische Gestaltung der Spielelemente war zu schlecht, so dass diese nicht erkennbar waren. Generell wurde aber die Spielidee von allen Teilnehmern als gut wahrgenommen. Es kamen die Vorschläge, dass das Spielkonzept weiter abstrahiert und vereinfacht werden solle. Des Weiteren soll der Linienbus nicht vom Spieler eingesetzt werden, da dieser kein Gegner sein soll.

Infotexte

Da die Spieler unterschiedliche Erfahrungen mit Spielen auf dem Smartphone haben werden, sollten Hilfetexte optional sein oder schnell „wegklickbar“ sein. Nach Ansicht der Experten ist ein einstellbarer Schwierigkeitsgrad zu komplex für eine Umsetzung. Ein levelbasierter Schwierigkeitsgrad, der ansteigt mit den Leveln ist einfacher umsetzbar, als eine dynamische Anpassung der Schwierigkeit im Spiel.



Erkennbarkeit

Die wichtigen Spielelemente wie zum Beispiel Fahrgäste müssen klar erkennbar sein. In dem ersten Konzept gab es Säulen als Fahrgäste, diese wurden von den Fokusgruppenteilnehmer nicht erkannt und waren zu abstrakt. Generell dürfen wichtige Spielelemente nicht zu abstrakt sein, da dies eine schnelle Erlernbarkeit behindert.

Rufbus

Die Bedienzeit im Spiel muss nicht der tatsächlichen Bedienzeit des Rufbusses entsprechen. Bei diesen Punkt waren sich alle Teilnehmer einig. Für ein spannenderes Spiel darf die Spielzeit schneller ablaufen als in der Realität.

Preissystem

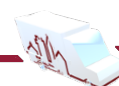
Ein mögliches Preissystem ist ein komplexes Spielelement, welches den Spielspaß erhöht, da eine komplexere Strategie notwendig ist (von den Teilnehmer als „Tycoon-Spiel-Prinzip“ bezeichnet). Die absolute Werte der Preise sind nicht relevant und müssen nicht mit der Realität übereinstimmen. Die Preise von einem möglichen Preissystem sollten vom Verhältnis her realistisch sein. Die Diskussion über manipulative Preise kam zu dem Ergebnis, dass diese nicht manipulieren dürfen, da diese die Glaubwürdigkeit des Serious Games senken.

Fahrgäste

Wenn der Fahrgast zu lange wartet, wäre es gut, wenn dieser irgendwann die Busstation verlässt und verschwindet. Eine Auswertung der durchschnittlichen Wartezeit ist ebenfalls möglich. Fahrgäste dürfen nicht nur direkt bei den Haltestellen stehen, sondern sollen, nach Ansicht der Verkehrsexperten, zu den Haltestellen hinlaufen.

3.2.8 Bewertung der Fokusgruppenmethode

Die Fokusgruppenmethode kann in diesem Fall insgesamt gesehen positiv bewertet werden. Sie hat erfolgreich viele Ideen und Verbesserungsvorschläge für das Spielkonzept hervorgebracht. Bei der Durchführung hat sich gezeigt, dass sich alle Teilnehmer erfolgreich eingebracht haben und die Teilnehmer homogen ausgewählt waren. Anders als in der Literatur (Schulz u. a.,



2012) beschrieben, haben sich alle Teilnehmer mit ähnlichen Redeanteilen eingebracht und haben dabei die eigenen Meinungen vertreten. Durch die Gruppenarbeiten im ersten Teil sind viele kreative Ideen entstanden, die bei zukünftigen Weiterentwicklungen berücksichtigt werden können. Der zweite Teil hat gezeigt, dass das bisherige Konzept vereinfacht werden muss. Die gezeigten Spielelemente wurden zum Teil nicht erkannt. Die aufgenommenen Meinungen sind zwar subjektiv, geben jedoch eine gute Einschätzung zu dem bisherigen Stand des Spielkonzepts wieder.

3.3 Iteration 3 - Heuristische Evaluation

Die erste Usability-Evaluation in dem angepassten Vorgehensmodell ist die Heuristische Evaluation. Dieses Kapitel erläutert zuerst die Grundlagen, um anschließend die Durchführung und die Ergebnisse aufzulisten.

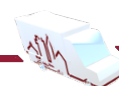
3.3.1 Methode

Usability

Die eingesetzten Evaluationen haben das Ziel eine möglichst gute Usability für B.u.S. zu erreichen. Der Begriff Usability beschreibt die Benutzbarkeit eines Systems und kann als Gütekriterium für die Gestaltung einer Benutzeroberfläche angesehen werden (Richter & Flückinger, 2016). Der Begriff ist in der Richtlinie ISO 9241-11 definiert als „das Ausmaß, in dem ein Produkt durch bestimmte Benutzer in einem bestimmten Nutzungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen“ (International Organization for Standardization, 2011). Durch die Breite des beschriebenen Begriffs kann die Usability nicht direkt gemessen werden.

Usability Tests

Usability Tests werden benutzt, um die Interaktionen von Testern mit der Benutzeroberfläche eines Systems zu evaluieren. Der Tester ist ein Benutzer, der bestimmte Aufgabe durchführen muss. Das Evaluieren kann nach verschiedenen Faktoren geschehen, wie zum Beispiel Design, Erlernbarkeit, Effizienz, Benutzerzufriedenheit oder Fehler (Rosenzweig, 2015). Die verschiedenen Tester müssen das Produkt unabhängig voneinander testen, um Probleme selber zu



entdecken und lösen zu können. Eine Beeinflussung durch den Testleiter sollte vermieden werden.

Es wird eine große Anzahl an Testpersonen benötigt um alle möglichen Usability-Probleme aufzudecken. Durch die serielle Durchführung ist eine Evaluation mit einer großen Anzahl an Testpersonen zeitaufwendig. Nach Nielsen (2000) kann eine Usability-Studie mit 5 Teilnehmern bereits bis zu 85% aller Usability-Probleme aufdecken. Das Diagramm 3.6 zeigt eine mögliche Näherung für die Aufdeckung von Usabilityproblemen in Abhängigkeit von den Testteilnehmern. Dazu wurde die folgende Näherungsformel aufgestellt: Gefundene Usability Probleme $P = N(1 - (1 - L)^n)$.

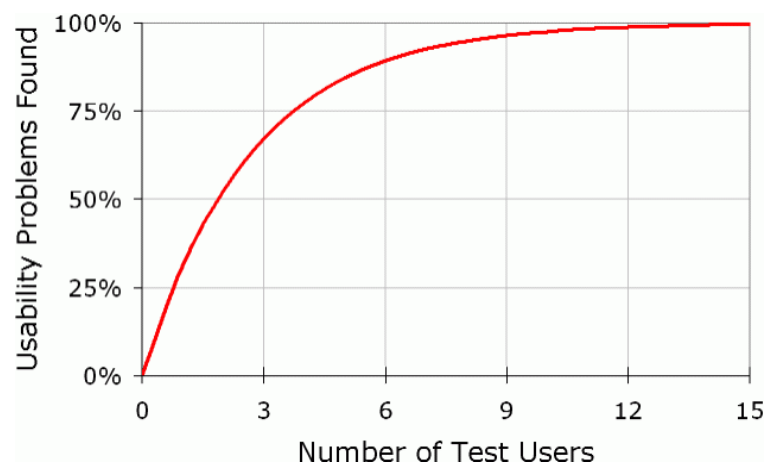
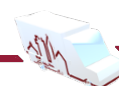


Abbildung 3.6: Anzahl der gefundenen Probleme nach Nielsen (2000)

Die Näherung geht darauf zurück, dass jeder neu hinzugewonnene Tester zwar neue Probleme aufdecken kann, allerdings auch die bekannte Probleme wiederholt. Nach Nielsen können mehrere getrennte Studien mit einer kleinen Teilnehmeranzahl demnach mehr Fehler aufdecken als eine große Studie.

Das Ziel von Usability-Tests im Rahmen dieser Arbeit ist es, Usability-Probleme zu finden. Diese zeigen die Schwächen des Spieldesigns und dienen der weiteren Entwicklung. Neben Usability-Problemen können mit diesen Tests andere Messgrößen wie zum Beispiel Erfolg oder benötigte Ausführungszeiten gemessen werden. Diese Messgrößen sind für die Entwicklung des Serious Games vorerst nicht relevant, deshalb werden diese an dieser Stelle nicht weiter betrachtet.



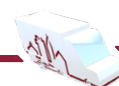
Heuristische Evaluation

Die Heuristische Evaluation ist eine Untersuchung der Usability eines Produkts in Hinblick auf Usability Prinzipien, den sogenannten Heuristiken (Nielsen, 1995a). Ein Beispiel für eine Heuristik ist die „Sichtbarkeit des Systemstatus“. Die Methode wurde 1990 von den beiden Usability-Experten Jakob Nielsen und Rolf Molich zum ersten Mal vorgestellt (Nielsen, 1995a). Diese Art der Evaluierung sieht vor, dass eine geringe Anzahl von Experten ein System einzeln untersuchen, um unabhängige Ergebnisse zu gewährleisten. Die Experten nutzen dabei das explorative Testen um Probleme zu finden. Jedes gefundene Problem wird mindestens einer Heuristik und einen Schweregrad zugeordnet.

Heuristiken

Heuristiken sind Usability Prinzipien, die die Software erfüllen muss. Die Heuristiken können bei der Vorbereitung der Evaluation auf die Testsituation und das Testsystem angepasst werden. Die benutzten Heuristiken für die Evaluation von B.u.S. basieren auf den zehn von Nielsen und Molich vorgestellten Heuristiken (siehe auch Nielsen, 1995a):

1. *Systemstatus* - Das System sollte dem Nutzer jederzeit sinnvoll Auskunft über den eigenen Status geben können.
2. *Übereinstimmung* zwischen realer Welt und Systemwelt - Die Sprache des Systems sollte den Nutzergruppen angepasst sein, Abläufe sollten eine logische Reihenfolge haben.
3. *Nutzerkontrolle* - Das System soll die Benutzer leiten und eine Fehlbedienung vermeiden.
4. *Konsistenz* - Für ähnliche Situationen sollten ähnliche Wörter verwendet werden, Verwendung von Standards.
5. *Fehlerverhütung* - Das System sollte Fehler vermeiden, der Nutzer sollte möglichst wenige Fehlermeldungen bekommen.
6. *Wiedererkennen* - Intuitives Erkennen der Spielelemente sollte möglich sein, so dass der Benutzer keine ausführliche Anleitung benötigt.
7. *Flexibilität* - Das System sollte auf Benutzer mit verschiedenen Lernraten eingestellt sein.



8. *Ästhetik und minimalistisches Design* - Das Design sollte auf die Aufgaben abgestimmt sein und keine überflüssiges Elemente enthalten.
9. *Hilfe bei der Fehlerbehebung* bzw. —erkennung und —diagnose - Alle Fehlermeldungen sollten direkt lesbar sein, Hilfsfunktionen sollten auf die jeweilige Aufgabe angepasst sein.
10. *Hilfe und Dokumentation* - Eine Dokumentation der Software und des Systems sollte ausführlich genug und übersichtlich aufgebaut sein.

Für die heuristische Evaluation des Serious Games gibt es angepasste Heuristiken. Da für das Serious Game keine öffentliche externe Dokumentation des Spiels vorgesehen ist und es außerdem keine Fehlermeldungen bei der Bedienung gibt, enthält die Evaluation für B.u.S. nicht die Heuristiken *Hilfe bei der Fehlerbehebung* (9.) und *Hilfe und Dokumentation* (10.). Stattdessen gibt es nur eine Heuristik mit dem Titel *Hilfe und Funktion*. Die Heuristik *Ästhetik und minimalistisches Design* (8.) hat bei einem Prototypen wie das Serious Game eine geringe Priorität und wird deshalb ebenfalls nicht verwendet.

Zusätzlich zu den bekannten Heuristiken wurden zwei neue Heuristiken eingeführt:

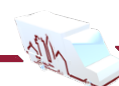
- *Lernförderlichkeit* - Das Serious Game sollte so gestaltet sein, dass es das Verständnis erhöht und nachhaltiges Lernen fördert.
- *Sonstiges* - Zusätzliche Heuristik für Probleme die keiner anderen Heuristik zugeordnet werden können.

Diese beiden Heuristiken sind angepasst an das Testsetting und die Tester. Für alle Tester war die heuristische Evaluation unbekannt, deshalb gibt es neben den bekannten Heuristiken die Heuristik *Sonstiges*, damit zu jedem Problem eine Heuristik zuordbar ist.

Schweregrad

Der Schweregrad dient der Zuteilung der freien Entwicklungsressourcen zur Behebung der wichtigsten Probleme (Nielsen, 1995b). Mit dem Schweregrad kann eine Priorisierung der Probleme erfolgen, um zu beurteilen ob weitere Ressourcen für die Entwicklung notwendig sind. Wenn die Evaluation Usability Probleme mit einem hohen Schweregrad aufdeckt, kann das System noch nicht verteilt werden.

Nielsen (1995b) schlägt für den allgemeinen Schweregrad die folgende Skala vor:



- 0 - Keine Zustimmung für das Usability Problem
- 1 - Kosmetisches Problem - sollte nur gefixt werden, wenn genügend Entwicklungszeit übrig ist
- 2 - Kleines Usability Problem mit geringer Priorität
- 3 - Großes Usability Problem, hohe Priorität
- 4 - Usability Katastrophe - muss auf jeden Fall gefixt werden

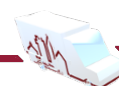
Zu jeder zugeteilten Heuristik müssen die Tester einen Schweregrad wählen. Die heuristische Evaluation für B.u.S. sieht für die gefundenen Probleme den Schweregrad 1 - 4 vor, da der Schweregrad 0 nur für die Bewertung fremder Probleme dient.

Heuristische Evaluationen sind ausgelegt auf die Problemfindung. Nicht jedes Problem wird von allen Testern gefunden, deshalb sind die Schweregrade nur Einzelwerte, und können auf Grund der geringen Stichprobe sehr subjektiv ausfallen. Nach Nielsen (1995b) ist der Mittelwert von mindestens drei Testern bei den meisten Projekten ausreichend für eine gute Aussage zur Usability.

3.3.2 Teilnehmer

Für die Heuristische Evaluation und den Thinking Aloud Test wurden Experten aus der Abteilung Human Factors vom DLR befragt, die das Grobkonzept aus dem Workshop bereits kannten. Insgesamt wurden fünf Personen befragt. Die Größe der Stichprobe wurde bewusst klein gehalten nach den Vorgaben von Nielsen (2000). Die Teilnehmer sind zwischen 28 und 33 Jahre alt, das Durchschnittsalter beträgt 29,2 Jahre (Standardabweichung 1,9 Jahre). Alle Teilnehmer sind als wissenschaftliche Mitarbeiter am DLR tätig und haben berufliche Erfahrung.

Die Erfahrung der Teilnehmer mit Smartphone-Spielen ist sehr unterschiedlich, 40% haben bis zu zwei Spiele auf dem Smartphone, 20% haben zwischen drei und fünf Spiele, 40% haben zwischen fünf und zehn Spiele auf dem eigenen Smartphone. Die Häufigkeit des Spielens, mit den Angaben täglich, wöchentlich und monatlich ist ähnlich verteilt wie die Verteilung der installierten Spiele. Die Fragebögen zu den demografischen Angaben sind im Anhang zu finden.



3.3.3 Durchführung und Testsetting

Der Entwicklungsstand zum Zeitpunkt der Durchführung der heuristischen Evaluation ist in Abbildung 3.7 zu sehen. Die Tester hatten zur Evaluation eine Webversion des Serious Games zur Verfügung. Dank der Plattformunabhängigkeit des benutzten Frameworks konnte die Webversion mit Hilfe von WebGL erstellt werden. Die Evaluation sah eine Konzentration auf dem Inhalt vor. Deshalb wurde die Webversion evaluiert und keine mobile App.

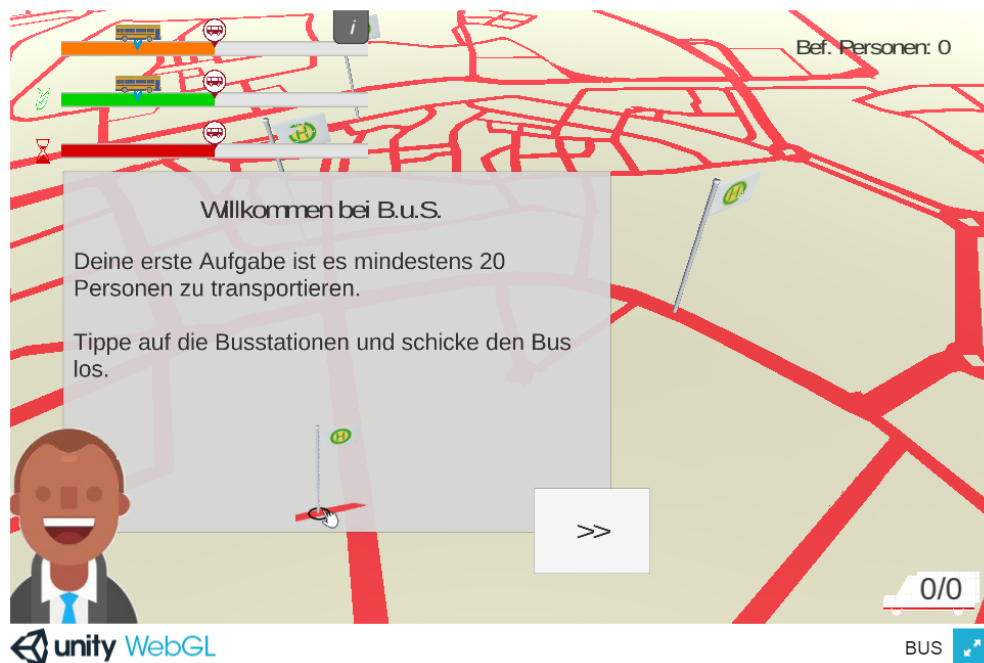
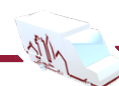


Abbildung 3.7: Entwicklungsstand zum Zeitpunkt der Heuristischen Evaluation

Die Abbildung zeigt die Startszene des Serious Games mit der ersten Anweisung. Die Anweisungen im Programm waren die einzigen Anweisungen der Tester bei der Evaluation. Auf weitere Anweisungen wurde bewusst verzichtet, da der spätere Nutzer der App ebenfalls keine weiteren Anweisungen vor dem ersten Start der App bekommt.

Als Testsetting wurden die Arbeitsplätze der Testpersonen ausgewählt, da diese eine Konzentration der Testpersonen auf das Serious Game erlauben. Die Abbildung 3.8 zeigt ein Beispielsetting. Auf dem Notebook ist das aktive Spiel zu sehen, dass zu Testzwecken von dem Tester erneut gestartet wurde. Der Zeitraum für die Evaluation wurde so gewählt, dass es keinen Zeitdruck für die Tester gab.



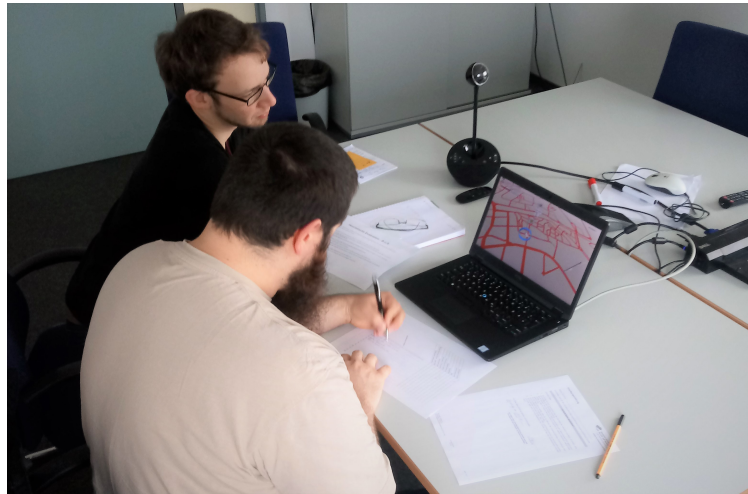


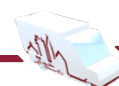
Abbildung 3.8: Testsetting bei der heuristischen Evaluation

3.3.4 Gefundene Probleme

Alle gefunden Probleme sind in den Tabellen 3.2 und 3.3 zu sehen, zusätzlich erläutert dieses Kapitel die Probleme mit den häufigsten Nennungen. Die den Problemen zugeordneten Heuristiken sind als Zahlen in der vierten Spalte zu finden. Die letzte Spalte in der Tabelle gibt den jeweils kleinsten und größten zugeordneten Schweregrad an. Für eine bessere Übersicht über die identifizierten Probleme sind diese eingeordnet in fünf Kategorien:

- (1) Benutzeroberfläche
- (2) Bedienung
- (3) Spielelemente
- (4) Spielkonzept
- (5) Hilfestellung

In der Kategorie Benutzeroberfläche (1) sind visuelle Probleme innerhalb der grafischen Benutzerschnittstelle gelistet. Die Tester haben das Spiel während des Tests mit der Maus bedient. Alle Probleme bei der Bedienung sind in der gleichnamigen Kategorie (2) gelistet. Innerhalb des Spiels sind die Probleme unterteilt in konzeptuelle Probleme (4) und Probleme mit einzelnen Spielelementen (3). Die einführenden Texte werden zwar in der Benutzeroberfläche dargestellt, für die Hilfestellung gibt es dennoch eine eigene Kategorie (5), da diese getrennt betrachtet werden können.



Das am häufigsten genannte Problem ist, dass die Ziele der im Rufbus befindlichen Fahrgäste unklar sind. Den meisten Testern war die Gesamtkapazität und die Restkapazität des Busses unklar. Dieses Problem haben vier von fünf Testern genannt. Am zweithäufigsten wurden bei der heuristischen Evaluation drei Probleme genannt. Das erste ist das Fehlen der Darstellung von Missionszielen, diese wurden nur in der Einführung genannt. Als zweites war die Bushaltestelle nur am Fuß anwählbar. Vier von fünf Testern wollten die Bushaltestellen aber an der Fahne mit dem Bushaltestellensymbol zuerst anwählen. Das dritte Problem mit drei Nennungen ist, dass die Route des Linienbusses im dritten Level unklar ist und am Anfang bei der eigenen Planung nicht mitgeplant werden kann. Eine visuelle Darstellung von der Buslinien gab es nicht, die Tester konnten die Linienführung erst nach einiger Zeit durch Beobachtung erfahren.

Mit zwei Nennungen gibt es insgesamt fünf gefundene Probleme. In den ersten Leveln gab es Anzeigen, die noch nicht benötigt und nicht erklärt wurden. Außerdem war in der Benutzeroberfläche die Schrift von der Zeit und den beförderten Personen zu klein und undeutlich dargestellt, da es keinen Hintergrund gab. Bei den Spielelementen gab es ein ähnliches Problem, die Sprechblasen waren auf Grund der Größe nur schlecht lesbar und haben sich überlappt. Bei den Hilfestellungen wurde zweimal bemängelt, dass die initialen Erklärungen nicht erneut aufrufbar waren.

Die gefundenen Usability-Probleme wurden Heuristiken zugeordnet, die wie folgt abgekürzt sind: 1 - Systemstatus, 2 - Übereinstimmung, 3 - Nutzerkontrolle, 4 - Konsistenz, 5 - Fehlerverhütung, 6 - Wiedererkennen, 7 - Flexibilität, 8 - Hilfe, 9 - Lernförderlichkeit, 10 - Sonstiges.

Die letzte Spalte gibt die zugeordneten Schweregrade an, hier angegeben mit jeweils kleinster und größter zugeordneter Schweregrad.

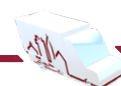


Tabelle 3.2: Identifizierte Probleme mit Heuristiken - Benutzeroberfläche, Bedienung, Spielelemente

Nummer	Heuristisch	Identifiziertes Problem	Heuristiken	Schweregrad
Benutzeroberfläche				
1	3	Missionsziele sind nicht sichtbar	1,5,10	1-3
2	2	Unnötige Anzeigen (Balken) in den ersten Leveln / Funktion der Balken ist nicht bekannt	7,10	2-3
3	2	Zeit und Beförderte Personen sind zu klein/undeutlich dargestellt	1,8	1
4	1	GUI-Elemente überdecken Spielelemente	10	1
5	1	Titel hat eine andere Schriftart	10	1
6	1	Näherung der negativen Spielziele könnten besser angezeigt werden	1	1
Bedienung				
8	3	Bushaltestelle ist nur am Fuß anwählbar (nicht an der Fahne)	3,5,10	2-3
9	1	Auswahl mehrerer Haltestellen ist nicht möglich	1	2
10	1	Kamera kann das Spielfeld verlassen	1	2
Spielelemente				
12	4	Inhalt (Farben der Fahrgäste) des Busses ist unklar	1,6,10	2-4
13	3	Anzahl und Kapazität des Busses ist unklar	1,3,5,8,9	2-3
14	3	Route des Linienbus ist unklar	1,2,5,8	2-3
15	1	Linienbushaltestellen nicht klar erkenn- & abgrenzbar	7,10	3-4
16	2	Sprechblasen der Spielelemente sind schlecht lesbar oder überlappen sich	1	1

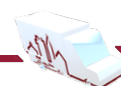
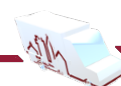


Tabelle 3.3: Identifizierte Probleme mit Heuristiken - Spielkonzept, Hilfestellung

Nummer	Heuristisch	Identifiziertes Problem	Heuristiken	Schweregrad
Spielkonzept				
17	1	Es gibt keine Bestrafung für zu lange Wartezeiten der Fahrgäste	10	1
18	1	Fahrgeschwindigkeiten sind sehr hoch	2	2
19	1	Ziele der verschiedenen Fahrgäste sind nicht sichtbar	1,2,10	2-4
Hilfestellung				
21	1	Erläuterungstexte sind ungenau (z.B. Erklärung der Balken)	8	3
22	2	Erklärungen sind nicht erneut aufrufbar	3,4,5,9,10	2-4
23	1	Fehlende „Werbetexte“ für die Qualität und Bedarf des Rufbusses	8,9	3-4
24	1	Begriffe nicht allgemein verständlich (z.B. On-Demand)	2	2
25	1	Nach dem letzten Erklärungsfenster startet sofort das Spiel	1	3



3.3.5 Auswertung der Heuristiken und der Schweregrade

Die Vergabe der Heuristiken bei den gefundenen Problemen ist sehr unterschiedlich. Es gibt kein Problem mit mehreren Nennungen, das nur eine zugeordnete Heuristik hat. Zu zwei Problemen wurden sogar fünf verschiedene Heuristiken zugeordnet.

Die Heuristiken können beim Kategorisieren der Probleme helfen und zeigen den Bereich des Programms, der die meisten Probleme hat. Die häufigste benutzte Heuristik ist (1) *System-status*, mit 12 Zuordnungen. Dies zeigt, dass der Systemstatus die meisten Probleme bei dem Serious Game aufweist. Die zweithäufigste benutzte Heuristik ist (10) *Sonstiges*, da diese zu unterschiedlichen Problemen zugeordnet werden konnte.

Die Vergabe der Schweregrade ist ebenfalls nicht eindeutig. Fünf Probleme haben den höchsten Schweregrad „Usability Katastrophe“ zugeordnet bekommen. Allerdings haben drei von den fünf Problemen nur eine einzige Nennung durch einen Tester. Die Anzahl der Nennungen werten diesen Schweregrad ab. Anhand der Nennungen und der Schweregrade wurden die Maßnahmen priorisiert.

3.3.6 Maßnahmen

Die beschlossenen Maßnahmen mit der Priorisierung sind in der Tabelle 3.4 zu finden. Jede Maßnahme ist mit einer Priorisierung zwischen Prio 1 (höchste Priorisierung) bis Prio 3 (niedrigste Priorisierung) priorisiert. Die erste Spalte der Tabelle ordnet die Maßnahmen den Problemen zu. Insgesamt gibt es vier Maßnahmen mit der Prio 1, fünf Maßnahmen mit der Prio 2 und sechs Maßnahmen mit der Prio 3. Die Priorisierung bestimmte die Reihenfolge der Implementierung. Das heißt, die Darstellung der Missionziele, des Businhalts, der Linienbusroute und die Änderungen an den Balkenanzeige wurde zuerst implementiert.

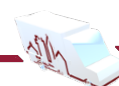
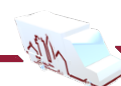


Tabelle 3.4: Maßnahmen zur Verbesserung der Usability

Zuteilung	Mögliche Lösung	Priorisierung
Benutzeroberfläche		
1	Missionsziele darstellen bei den Elementen	Prio 1
2	Balkenanzeigen ausblenden, wenn nicht benötigt	Prio 1
3	Schrift vergrößern, Hintergrund ändern	Prio 2
5	Schriftart des Titels überprüfen	Prio 3
6	Aufblinken der Elemente bei Näherung des Spielziels	Prio 3
Bedienung		
8	Fahne als klickbares Element	Prio 2
10	Kameravariablen begrenzen	Prio 3
Spielelemente		
12,13	Businhalt (mit Anzahl und Kapazität) in der GUI darstellen	Prio 1
14	Linienbus vor dem Spielstart losfahren lassen und Marker für die nächste Bushaltestelle	Prio 1
15	Linienbushaltestellen später im Spiel einführen	Prio 2
16	Weniger und größere Sprechblasen darstellen und in Blickrichtung drehen	Prio 3
Spielkonzept		
17	Sterne als Bewertungssystem einfügen	Prio 3
Hilfestellung		
21	Erläuterungstexte vereinfachen und ggf. abändern	Prio 2
22	„Kleiner Berater“ für den erneuten Aufruf der Texte	Prio 2
23	Texte nach dem Spielende einfügen	Prio 3



3.4 Iteration 4 - Thinking Aloud Test

Thinking Aloud ist eine Forschungsmethode, bei der die Beteiligten die eigenen Gedanken laut aussprechen (Charters, 2003). Im Rahmen eines Thinking Aloud Tests kann diese Methode genutzt werden, um die Usability eines Systems zu evaluieren. Beim Thinking Aloud Test verbalisiert ein Tester seine Gedanken beim Testen der Benutzeroberfläche eines Systems. Der Test ist ein einfacher Usability-Test, der flexibel eingesetzt werden kann und schnell erlernbar ist. Die Methodik bietet den Vorteil, dass Usability-Probleme bereits während des Spiels identifiziert und dokumentiert werden können. Durch die direkte Dokumentation werden Erinnerungsverluste vermieden. Der Test sammelt Feedback, kann Risiken aufdecken, zeigt aber nicht alle vorhandenen Probleme (Nielsen, 2012).

Nach Nielsen (2012) besteht ein Thinking Aloud Test aus drei Schritten:

1. Repräsentative Benutzer auswählen
2. Test erläutern und Instruktionen verteilen
3. Benutzer sprechen lassen (Monolog) bei gleichzeitigen Test des Systems

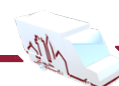
Thinking Aloud hat die größten Vorteile bei der einfachen und flexiblen Durchführung. Die Methode kann zu nahezu jedem Thema angewendet werden.

Ein größerer Nachteil ist, dass der Schweregrad der Probleme nur mit Hilfe einer Nachfrage, über die Aussprache oder über die Anzahl der Nennungen bestimmt werden kann.

3.4.1 Evaluationsdesign und Durchführung

Die Tester und Testsetting für den Thinking Aloud Test sind die gleichen, wie bei der Heuristischen Evaluation. Als Tester für die Evaluation wurde die Experten aus dem vorangegangenen Workshop und Evaluation ausgewählt, die das Spielkonzept bereits kennen. Als Testsetting wurden die Arbeitsplätze der Personen ausgewählt, da diese eine Konzentration der Testpersonen auf das Serious Game erlauben. Der Zeitraum für die Evaluation wurde so gewählt, dass es keinen Zeitdruck für die Tester gab.

Die Testerklärung fand verbal durch eine kurze Ansage statt. Den meisten Testern war die Methode schon bekannt. Instruktionen und Erläuterung in Papierform gab es für diesen Test nicht. Das Spiel und die Aufgaben des Spiels sollten die Tester alleine führen.



Die Abbildung 3.9 zeigt den Entwicklungsstand von B.u.S. zum Zeitpunkt des Thinking Aloud Tests. Deutlich zu sehen sind auf der Abbildung die Verbesserungen wie zum Beispiel der graue Hintergrund im oberen Bereich, die Inhaltanzeige der Busses unten rechts oder die Zielanzeige der Busse in Form von Ringen.

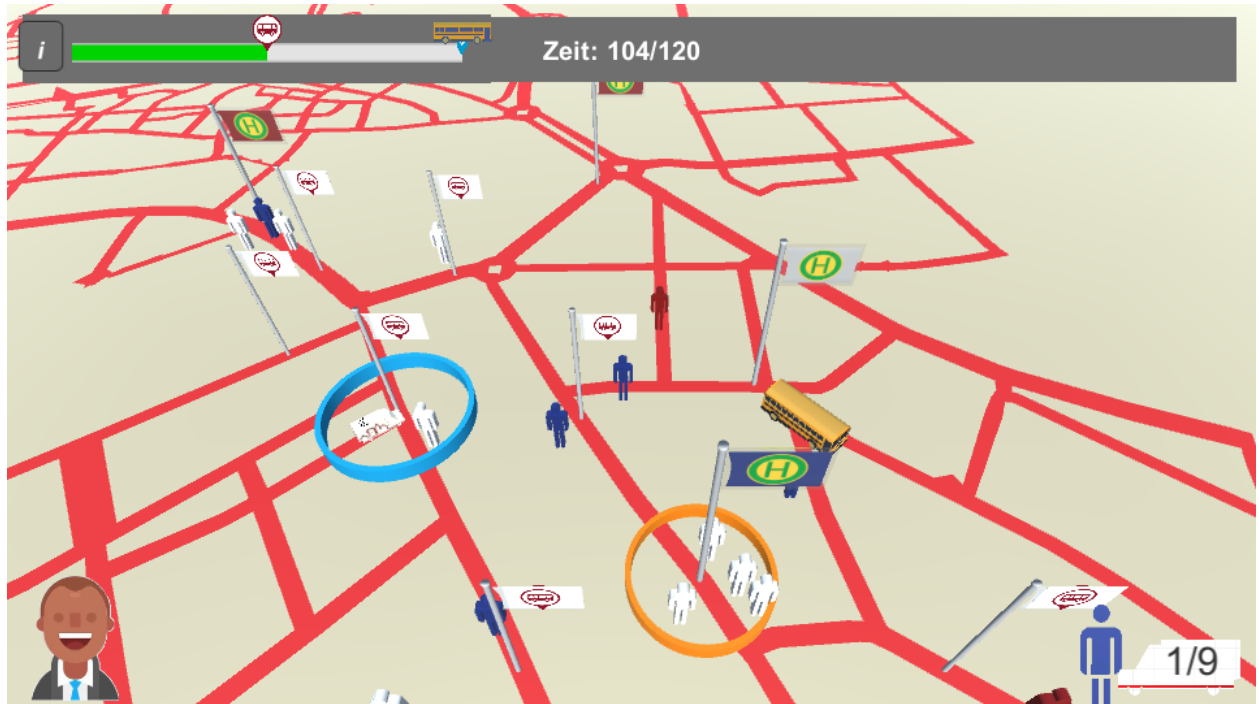


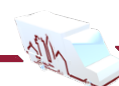
Abbildung 3.9: Entwicklungsstand: Ausschnitt aus dem Serious Game

3.4.2 Beobachtungen

Bei der Durchführung fiel auf, dass einige Tester nur wenige Gedanken verbalisieren, es entstanden zum Teil mehrere längere Pausen beim Redefluss. Durch offene Fragen wurde dieser Nachteil versucht zu reduzieren, konnte aber nicht vermieden werden.

Durch die direkte Beobachtung der Tester konnten Probleme weiter spezifiziert und Lösungsansätze aufgedeckt werden.

Die Durchführungen dauerte je nach Ausführlichkeit zwischen 30 und 45 Minuten pro Teilnehmer. Alle Thinking Aloud Monologe wurden mit Hilfe der Aufnahmefunktion eines Smartphones aufgezeichnet und sind im Anhang auf der beigelegten DVD zu finden.



3.4.3 Auswertung

In dem Diagramm in Abbildung 3.10 ist die Anzahl der gefundenen Probleme pro Tester (T1 - T5) aufgetragen. Der Balken „Gesamt“ gibt alle gefundenen Probleme ohne Mehrfachnennungen an.

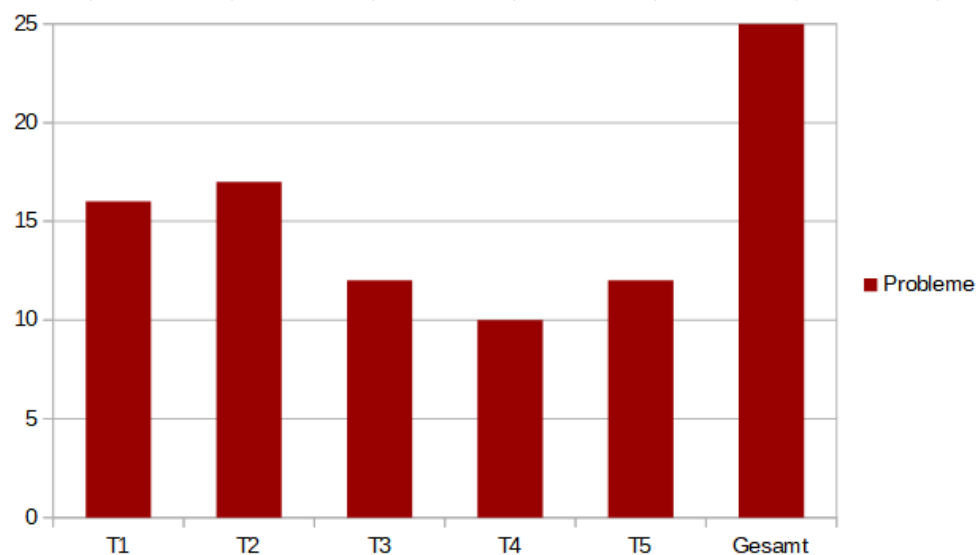
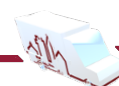


Abbildung 3.10: Nennungen der Probleme aufgeteilt nach Testern (T1 - T5)

Die zweite Usability-Evaluation von B.u.S. hat insgesamt 25 verschiedene Usability-Probleme ergeben. Zur Auswertung wurden diese Probleme erneut in die fünf Kategorien *Benutzeroberfläche*, *Bedienung*, *Spielelemente*, *Spielkonzept* und *Hilfestellung* eingeordnet. Eine Auswertung als Balkendiagramm zeigt die Abbildung 3.10. Die einzelnen Tester haben bei der Evaluation jeweils zwischen 10 und 17 Probleme gefunden. Im Durchschnitt hat ein Tester 13,4 Probleme gefunden.

Das gefundene Problem mit der Nummer 25 (Erneutes Auftreten eines Game Over nach einem ersten Game Over) ist allerdings nicht zu den vorhandenen Kategorien einordbar, deshalb hat dieses die neue Kategorie *Softwarefehler*. Die geringe Anzahl an Softwarefehlern zeigt, dass weitere Softwaretests zur Qualitätsverbesserung notwendig sind und dass einfaches exploratives Testen im Rahmen einer Evaluation nicht alle Softwarefehler aufdeckt.

Es gibt insgesamt fünf Probleme mit fünf Nennungen, drei Probleme mit vier Nennungen, drei Probleme mit drei Nennungen, sieben Probleme mit zwei Nennungen und sieben Probleme mit nur einer Nennung. Nachfolgend werden nur die Probleme erläutert, die von allen Testern gefunden wurden. Von allen Testern wurden die neu eingeführten Symbole für Stationen



außerhalb des Bildschirms nicht erkannt oder nicht entdeckt beim Spielen (7). Des Weiteren wurde von allen bemängelt, dass die Stationen zum Teil zu nahe beieinander liegen (8), dass bei einigen Leveln Stationen außerhalb des Startbildschirms liegen (15/16) und dass die Route des Linienbusses nach der Einführung des Linienbusses unklar ist (17).

Bei der weiteren Auswertung wurde erneut zu allen Problemen Lösungsvorschläge definiert und diese für eine Umsetzung priorisiert. Die Priorisierung erfolgte nach den Anzahl der Nennungen und nach der Relevanz der Probleme. So haben die oben aufgelisteten Probleme mit deren Lösungen die höchste Priorität erhalten.

Das Spielkonzept an sich wurde beim Test gelobt, die Änderungen gegenüber der letzten Testversion wurden von den Testern ebenfalls positiv hervorgehoben. Die Schweregrade der gefundenen Probleme sind im Vergleich zur letzten Evaluation kleiner geworden. Dies zeigt das Vorgehensmodell mit der Dreiecksform (siehe Abbildung ??). Alle gefundenen Probleme, einschließlich der oben genannten Probleme, lassen sich mit kleineren Änderungen der Software beheben oder verbessern.

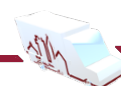
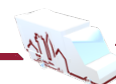


Tabelle 3.5: Identifizierte Probleme beim Thinking Aloud Test

Nummer	Nennungen	Identifiziertes Problem
Benutzeroberfläche		
1	2	Personengruppen-Symbol ist unklar, Bild schlecht erkennbar
2	1	Roter Balken (Wartezeit) ist unklar
3	1	Balken verdeckt die Info, wenn zwei Balken vorhanden sind
4	1	Voller grüner Balken bildet einen schlechten Wert ab
Bedienung		
5	1	Möglichkeit des Zoom ist unbekannt
6	1	genauer Start nach den Dialogen ist unklar (kein getrennter Button)
Spielelemente		
7	5	Symbole für die Haltestellen am Bildrand nicht erkannt
8	5	Stationen z.T. nahe beieinander, Fahrgäste sind nicht klar den Stationen zuordbar
9	4	Berechnung der Umweltfreundlichkeit ist unklar
10	4	Längere Wartezeiten unter den Fahrgästen sind nicht erkenntlich
11	3	Personen steigen nicht ein, obwohl der Bus an der Haltestelle steht
12	2	Zahlen sind auf dem Rufbus nicht erkennbar
13	2	Farben der Flaggen der Linienbushaltestellen sind nicht klar erkennbar
14	2	Bus fährt nicht den direkten Weg an einer Stelle
Spielkonzept		
15	5	Zoom erforderlich für die Anzeige aller Haltestellen
16	5	Tester gehen davon aus, dass alle Stationen im Bild sind, ist aber nicht der Fall
17	5	Route des Linienbusses ist unklar
18	4	Unklarheit bezüglich der Stern-Bewertung nach dem Levelende
19	3	lange Wartezeit am Anfang bis zu den ersten Fahrgästen
20	2	Personengruppen-Symbol ist nicht auf dem Boden, sondern schwebt in der Luft
21	2	Schwierig alle Fahrgäste mitzunehmen
22	1	Grund des Verlierens unklar
Hilfestellung		
23	2	Es fehlen weitere Erklärungen für die Konzepterklärung
24	1	Position und Einheit der Zeitanzeige wird nicht beschrieben
Softwarefehler		
25	3	GameOver direkt nach einem ersten GameOver ohne Erreichen der Zielwerte

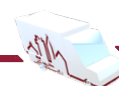


3.5 Iteration 5 - Softwaretests

Die Softwaretests bilden den letzten Teil des Vorgehensmodells vor der Übergabe der Software. Nach der Übergabe erfolgt das Verteilen, Betreiben und Warten der Software. Diese Schritte bildet das Modell nicht ab und sind nicht Bestandteile dieser Arbeit. Das Thema Softwaretest ist sehr umfangreich und kann in mehrere Teststufen unterteilt werden. Deshalb gibt dieses Kapitel nur einen kurzen Überblick über den Testbegriff.

Der Test einer Software ist nach Spillner & Linz (2012) die Ausführung eines Testobjekts unter spezifizierten Bedingungen. Ein Testobjekt ist in der Regel ein bestimmter Teil der Software. Die Softwaretests haben zum Ziel Fehlerwirkungen nachzuweisen, die Qualität einer Software zu bestimmen, das Vertrauen in das Programm erhöhen und Fehlerwirkungen vorzubeugen. Ein Fehler ist dabei das Nichterfüllen einer festgelegten Anforderung und macht sich durch eine Fehlerauswirkung bemerkbar. (Spillner & Linz, 2012)

Jede Software muss während der Entwicklung getestet werden, da es beim Entwickeln einer umfangreichen Software immer Ausnahmesituationen gibt, die der Entwickler nicht eingeplant hat. Deshalb ist keine Software komplett fehlerfrei. (Spillner & Linz, 2012)



4 Projektmanagement und Konzeptrealisierung

Das folgende Kapitel erläutert die Konzeptrealisierung und Umsetzung des Serious Games. Dabei wird zuerst auf das Projektcontrolling, die agile Entwicklung und die benutzten Systeme eingegangen. Anschließend werden die Umsetzungen der wichtigsten Spielelemente mit deren didaktischen Bedeutungen erläutert.

4.1 Projektcontrolling

Zu jeder Evaluation gab es eine Post-Evaluation, bei der die wichtigsten demografischen Daten der Teilnehmer abgefragt wurden. Im Vorhinein gab es für die Verarbeitung der Daten eine Einwilligungserklärung. Mit den Post-Evaluationen wurde die Projektqualität überprüft. Dazu haben die Teilnehmer jeweils eine Note (einfaches Schulnotensystem) für die Spielidee und für die Umsetzung gegeben. Bei beiden Evaluationen hat die Spielidee eine Note von 1,6 mit einer Standardabweichung von 1,1 erhalten. Die Umsetzung hat sich dabei von 2,6 auf 2,0 (Standardabweichung 0,9 bzw. 1,1) verbessert. Demzufolge hat sich nach den subjektiven Empfinden der Teilnehmer die Qualität der Software während des Projekts gesteigert.

4.2 Anforderungsmanagement

Das Anforderungsmanagement hatte die Aufgabe, die funktionalen und nichtfunktionale Anforderungen von B.u.S. zu verwalten und zu dokumentieren. Die Verwaltung der Anforderungen erfolgte, wie die Softwareentwicklung auch, nach einem agilen Ansatz. Die agile

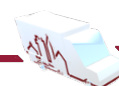
Softwareentwicklung basiert auf dem sogenannten agilen Manifest. Das agile Manifest, welches 2001 von 17 Personen unterschrieben und publiziert wurde, hat vier Kernaussagen (Beck, 2001):

- **Individuen und Interaktionen** stehen über Prozesse und Systeme
- **Funktionierende Software** ist wichtiger als eine umfassende Dokumentation
- **Zusammenarbeit mit den Kunden** steht über Vertragsverhandlungen
- **Reagieren auf Veränderungen** ist wichtiger als das Befolgen eines Plans

Jedes Vorgehensmodell, dass auf diesen Aussagen aufbaut unterstützt die agile Entwicklung (Epping, 2011).

Für das agile Aufgaben- und Anforderungsmanagement wurde ein Kanban-Board benutzt. Kanban ist eine Methode, die der Japaner Taiichi Ohno für die Produktionsprozesssteuerung des Autoherstellers Toyota entwickelt hat (Epping, 2011). Bei der Methode wird ein sogenanntes Kanban-Board verwendet, das die Phasen der Wertschöpfungskette anzeigt. Bei dem Aufgabenmanagement bei der Softwareentwicklung kann ein solches Board den Status der Umsetzung der Aufgaben zeigen.

Die Abbildung 4.1 zeigt das verwendete digitale Kanban-Board. Für das Kanban-Board wurde die webbasierte Software Trello verwendet. Die Abbildung zeigt das Board zu Beginn der dritten Iteration. Es ist eingeteilt in drei Bereiche: Offene Anforderungen, in Bearbeitung und abgeschlossene Anforderungen. Zusätzlich gibt es ein Archiv für Ideen oder verworfene Anforderungen, welches nicht sichtbar auf der Abbildung ist. Jede Anforderung ist genau einer Iteration zugeordnet.



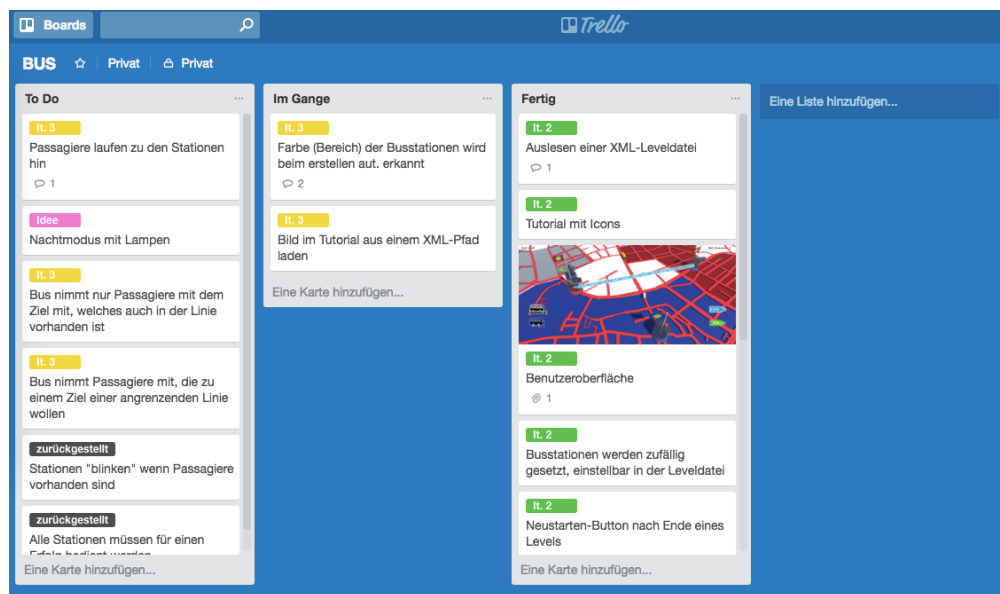
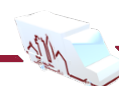


Abbildung 4.1: Kanban-Board in Trello

4.3 Software-Framework

Das Serious Game B.u.S. nutzt als Framework die Spiel-Engine Unity. Unity ist eine plattformunabhängige Spiel-Engine für Videospiele mit 2D- oder 3D-Grafik (Unity Technologies, o.D.). Eine Spiel-Engine ist ein Software-Framework für Computerspiele zur Steuerung der Darstellung und der Logik. Neben der Grafik ist eine Spiel-Engine für Physik, Sound, Benutzeroberfläche und Datenverwaltung zuständig.

Unity ist plattformunabhängig. Das heißt, Zielplattformen können Smartphones, Computer, Consolen und VR/AR-Brillen sein. Die Entwicklung erfolgt für alle Plattformen mit der Programmiersprache C#. Als weiteren Bestandteil gibt es eine eigene Entwicklungsumgebung. Diese kann die aktuelle Szene darstellen und erlaubt ein direktes Testen ohne Exportierung. Alle Spielobjekte wie Modelle, Kamera oder Licht sind in einem Szenengraph angeordnet. Die Entwicklungsumgebung verwaltet alle Assets. Assets sind Komponenten die zur Spielentwicklung genutzt werden. Die Importierung der Assets erfolgt per Drag-&Drop. Jedem Spielobjekt kann ein C#-Skript zugeordnet werden. Die C#-Skripte bilden die Logik in dem Spiel ab. Die Entwicklung der C#-Skripte erfolgte in dieser Arbeit mit der Entwicklungsumgebung Microsoft Visual Studio.



4.4 Softwarearchitektur

Für die Softwarekomponenten nutzt Unity aggregierte Entity-Components (Prühs, 2014). Das heißt, dass jedem Spielobjekt beliebig viele Komponenten und Skripte zugeordnet sein können. Alle Skripte eines Spielobjekts erben von einer vorgegebenen Hauptklassen. Das bedeutet, dass die Spielobjekte auf dieser Hauptklasse aufbauen und deren Methoden nutzen können. Bei B.u.S. ist jedem Spielobjekten maximal ein Skript zugeordnet.

Die Abbildung 4.2 zeigt ein Klassendiagramm, dass die Softwarearchitektur von B.u.S. wiedergibt. Als zentrales Element nutzt die Architektur das Architekturmuster Blackboard. Das Architekturmuster Blackboard wird zur Speicherung und Verwaltung von Steuerungsdaten genutzt (Buschmann u. a., 1998). Das Blackboard bildet bei B.u.S. die Schnittstelle zwischen Benutzeroberfläche und der restlichen Software. Das Blackboard ist ein leeres Spielobjekt mit einer angehängten Klasse. Es dient ebenfalls zur Berechnung von Variablen, wie zum Beispiel Fahrzeugdaten, die anschließend für die Siegbedingungen ausgewertet werden. Die Auswertung der Siegbedingungen und die Steuerung des Spiels erfolgt durch die Klasse MyGameController. Die Benutzeroberflächen wird gesteuert durch den ButtonController und weiteren angehörigen Klassen. Alle Assoziationen¹, die in dem Klassendiagramm zu sehen sind, wurden manuell in der Entwicklungsumgebung als Referenz übergeben.

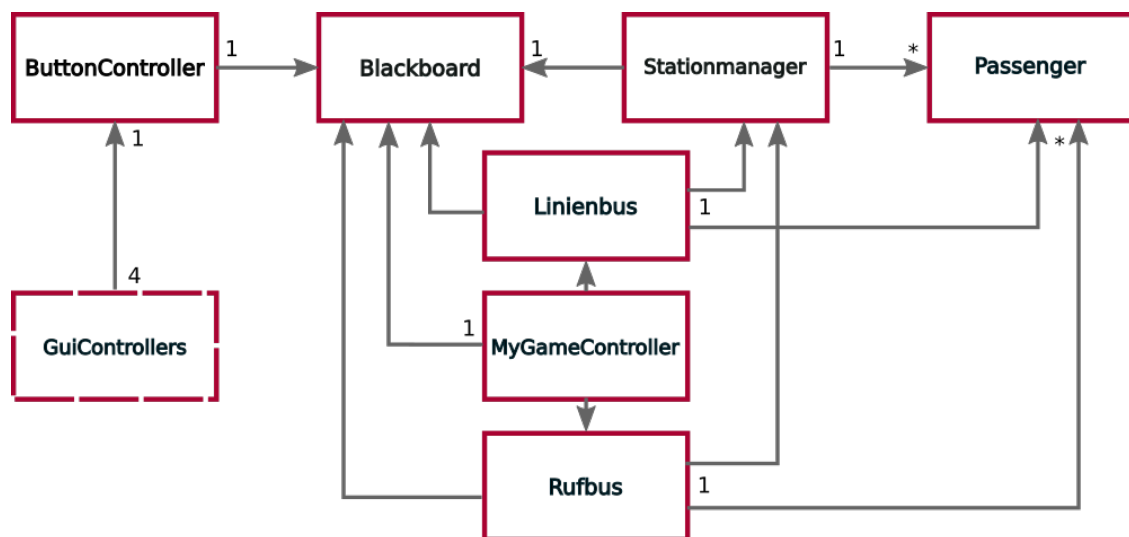
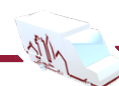


Abbildung 4.2: Softwarearchitektur von B.u.S.

¹Eine Assoziation beschreibt die Verbindung zwischen zwei Klassen und wird in dem Klassendiagramm durch eine Linie repräsentiert.



4.5 Spielablauf

Die Abbildung 4.3 zeigt den Spielablauf als Aktivitätsdiagramm. Jedes Level startet durch die Anzeige von Einführungstexten. Diese erläutern die Spielelemente und geben eine Einführung zu dem Thema Mobility-On-Demand. Erst wenn der Spieler diese gelesen hat, beginnt das Spiel. Durch die Auswahl einer Busstation fährt der Rufbus los. Der Spieler hat die Aufgabe die erscheinenden Fahrgäste zu transportieren. Das Erscheinen der Fahrgäste ist abhängig von einer Zufallsfunktion, das Level gibt dabei die Fahrgäste pro Minute für eine Haltestelle vor. Größere Gruppen von Fahrgästen werden einige Sekunden vorher mit Hilfe eines Symbols angekündigt. Das Generieren von größeren Fahrgastgruppen erfolgt ebenfalls nach einer Zufallsfunktion. Das Spiel endet mit Ablauf eines Timers oder mit dem Erreichen einer gewissen Fahrgastanzahl. Nach dem Spielende werden weitere Erläuterungstexte und eine Bewertung angezeigt.

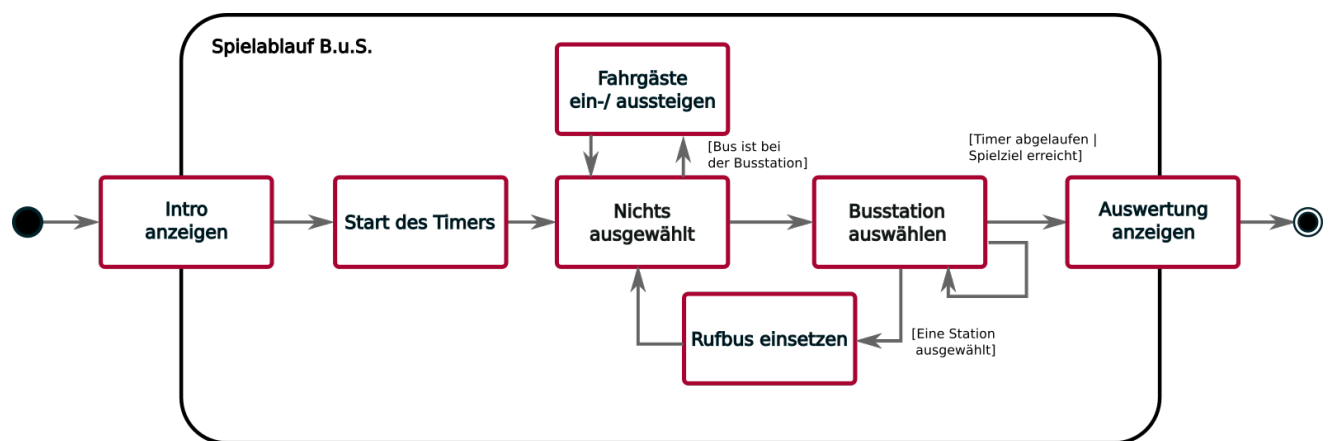
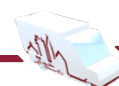


Abbildung 4.3: Aktivitätsdiagramm des Spielablaufs

4.6 Spielelemente

Dieses Kapitel gibt einen Überblick über das Design der wichtigsten Spielelemente von B.u.S und deren didaktische Bedeutung. Das fertige Serious Game ist als Android- und PC-Version im Anhang auf der beigelegten DVD zu finden.



Straßennetz

Die Grundlage des Spiels ist ein vereinfachtes Straßennetz von Schorndorf. Das Straßennetz basiert auf einer OpenStreetMap-Karte der Stadt. Allerdings wurden für das Spiel möglichst alle Einbahnstraßen und Fußwege entfernt, da diese Straßen von dem Rufbus nicht befahren werden. Die Abbildung 4.4 zeigt einen Ausschnitt vom Innenstadtbereich in der Entwicklungsumgebung. An einigen Stellen wurde die Anzahl der Haltestellen gegenüber der Realität verringert, da bei der realen Karte (siehe Abbildung 1.3) einige Haltestellen sehr nahe beieinander liegen. Die Linienbushaltestellen werden in B.u.S. mit Hilfe einer Fahne und dem allgemein bekannten gelb-grünen H-Symbol abgebildet. Die Rufbushaltestellen haben hingegen eine Fahne mit einem roten Bus-Symbol.

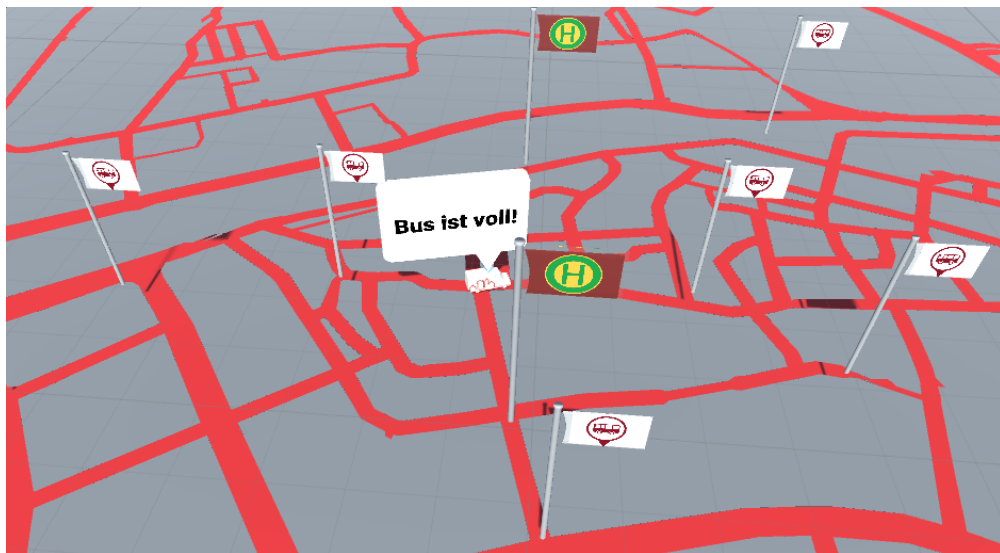
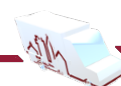


Abbildung 4.4: Rufbus- und Linienbushaltestellen im Spiel

Busse

In dem Spiel gibt es zwei verschiedene Busarten. Der Bedarfsbus wird als abstrakter Kleinbus dargestellt und hat das Logo des Reallabor Schorndorfs auf der rechten Fahrzeugseite. Das Aussehen ist angelehnt an die reale Lackierung der Kleinbusse.

Zudem gibt es einen Linienbus, für diesen wurde ein gelber Linienbus aus dem Unity-Asset-Shop ausgewählt. Der gelbe amerikanische Bus hat keine lokale Lackierung, das Design ist allerdings allgemein für ein Schul- und Linienbus bekannt.



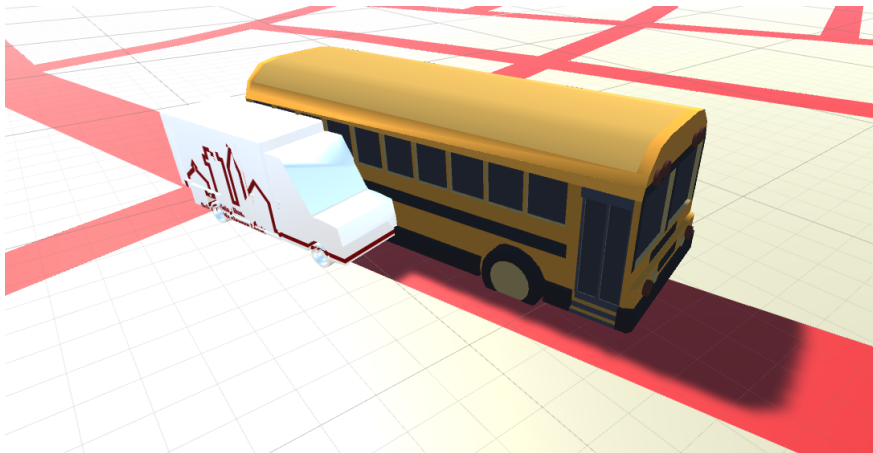
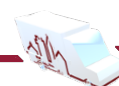


Abbildung 4.5: Busmodelle im Spiel

Fahrgäste

Das erste Konzept für Bus (siehe Abbildung 3.3) sah Säulen für die Darstellung der Fahrgäste vor. Bei der Durchführung des Workshops wurden diese von den Teilnehmer nicht als Fahrgäste erkannt. Deshalb wurde das Konzept geändert und das Serious Game benutzt in der aktuellen Version eingefärbte Ampelmännchen für die Darstellung der Fahrgäste. Die Abbildung 4.6 zeigt eine Spielszene mit Fahrgästen, die an Haltestellen warten. Das Fahrgastgruppensymbol oberhalb der linken Haltestelle kündigt eine größere Gruppe von Fahrgästen an. In der Abbildung ist ebenfalls die zweistufige Rückmeldung der Zufriedenheit der Fahrgäste zu sehen. Diese wird mit einem Smiley dargestellt und ist abhängig von der Wartezeit des Fahrgasts.



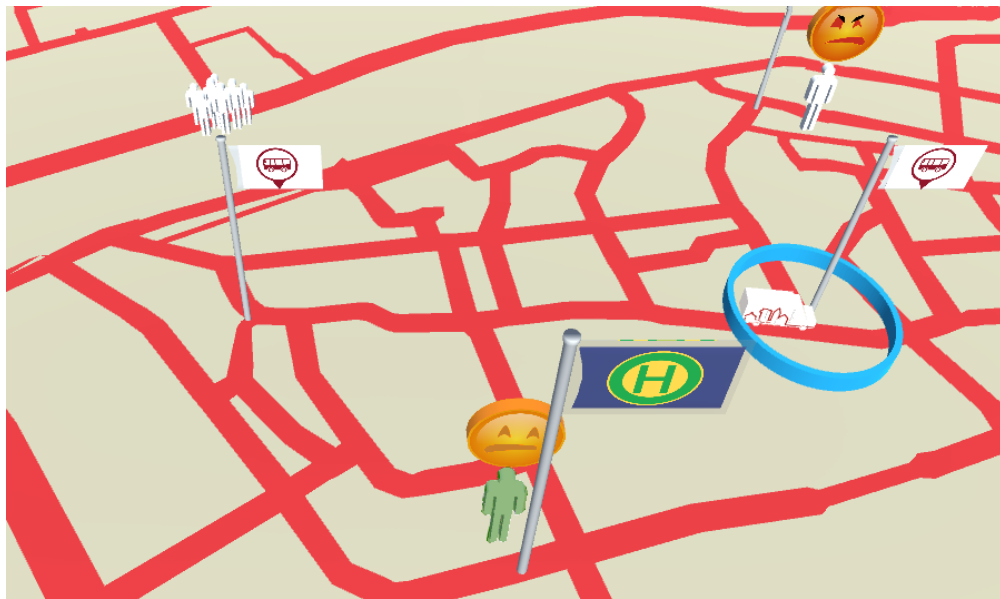


Abbildung 4.6: Fahrgäste im Spiel

Benutzeroberfläche

Das Serious Game hat eine minimalistisch gehaltene Benutzeroberfläche. Eine Überladung von Elementen wurde bewusst vermieden. Der Nutzer sieht in jedem Level nur die Elemente, die notwendig für das Level sind. Im ersten Level sieht der Nutzer im oberen Bereich des Spiels nur einen Zähler für die beförderten Fahrgäste. In späteren Levels sind auch die Zeit und Anzeigen für die Wartezeit oder Umweltfreundlichkeit zu sehen. Die Abbildung 4.7 zeigt die Infoleiste aus dem fünften Level mit dem Balken zur Darstellung der Umweltfreundlichkeit.



Abbildung 4.7: Infoleiste aus Level 5 mit Darstellung der Umweltfreundlichkeit

Einführungs- und Reflektionstexte

Der größte Teil der Konzeptvermittlung findet mit Hilfe von kurzen Texten statt. In B.u.S. gibt es Einführungstexte vor jedem Level. Ein Beispiel für einen Einführungstext ist in Abbildung 4.8 dargestellt. Diese Abbildung zeigt den ersten Text des ersten Levels mit einer

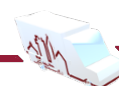


Abbildung eines realen Rufbusses. Die Texte sind bewusst kurz gehalten um den Spielspaß nicht zu gefährden. Die Einführungstexte können zu jedem Zeitpunkt in einem Level über den Berater erneut aufgerufen werden.

Nach dem Ende eines Levels werde Reflektionstexte und eine Sternbewertung, wie in Abbildung 4.9 dargestellt, eingeblendet. Zu jedem Level gibt es drei verschiedene Texte mit unterschiedlichen Inhalten und Hinweisen. Der Spielerfolg entscheidet welcher Text eingeblendet wird. Die Texte geben Hinweise für eine bessere Spielstrategie.

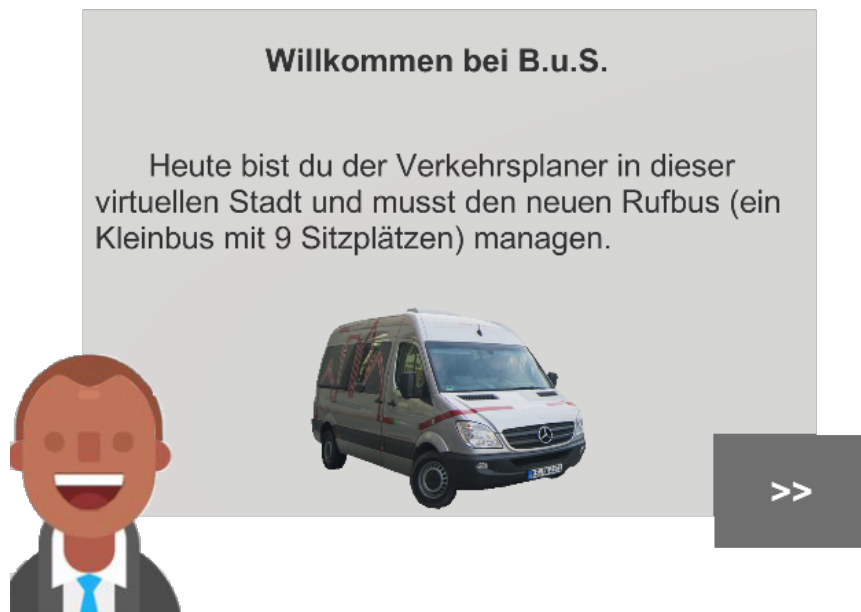
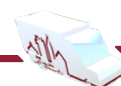


Abbildung 4.8: Einführungstext aus dem ersten Level



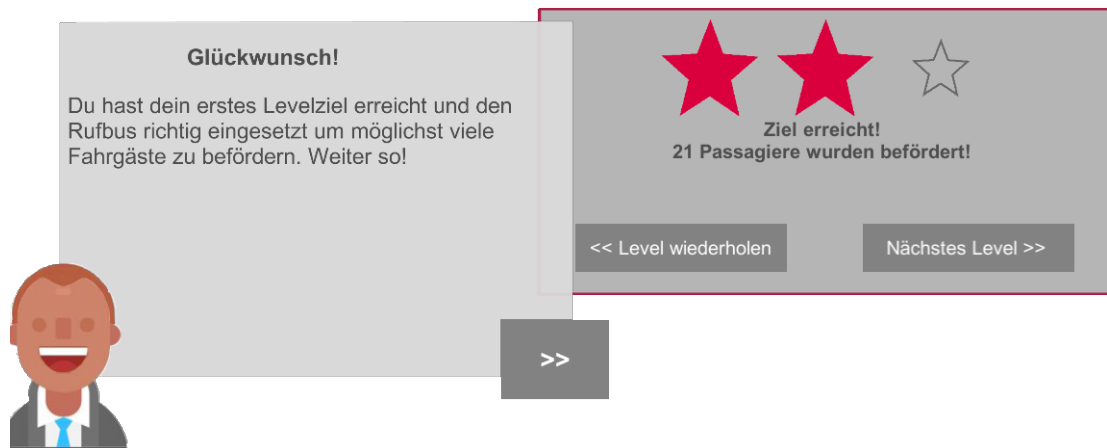
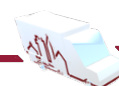


Abbildung 4.9: Reflektionstext nach dem Erreichen eines Spielziels

4.7 Spiellevel

Für die Speicherung der Spiellevel wird die Extensible Markup Language (XML) verwendet. Ein Beispiel für eine solche Datei ist im Anhang auf der beigelegte DVD zu finden. Die XML-Datei wird zur Laufzeit der Software eingelesen. Dies hat zum Vorteil, dass Level verändert werden können, ohne dass Änderungen im Quelltext notwendig sind. Da für eine Änderung kein Quelltext verändert werden muss, ist ein Softwareentwickler für die weitere Entwicklung der Level mit den bestehenden Spielelementen nicht notwendig.

Die Abbildung 4.10 zeigt den didaktischen Ablauf verteilt auf die Spiellevel. In den ersten Spielleveln werden die Grundlagen des Spiels erläutert. Das erste Level erläutert den Rufbus und das Grundprinzip des Spiels. Die nächsten beiden Level führen größere Personengruppen und Linienbushaltestellen ein. Die Linienbushaltestellen dienen als Fahrziele der Fahrgäste. Ab dem vierten Level gibt es einen Linienbus, der dem Spieler hilft und die Linienbushaltestellen bedient. Die Linienbushaltestellen haben ein größeres Fahrgastaufkommen als die Rufbushaltestellen. Spätere Level führen weitere Spielziele ein, die das Spiel in Form eines Balkens darstellt. Es gibt ein Spielziel für die maximale Wartezeit der Fahrgäste und ein weiteres Spielziel, dass die Umweltfreundlichkeit abbildet. Die Umweltfreundlichkeit wird gemessen über die Anzahl der Leerfahrten des Busses.



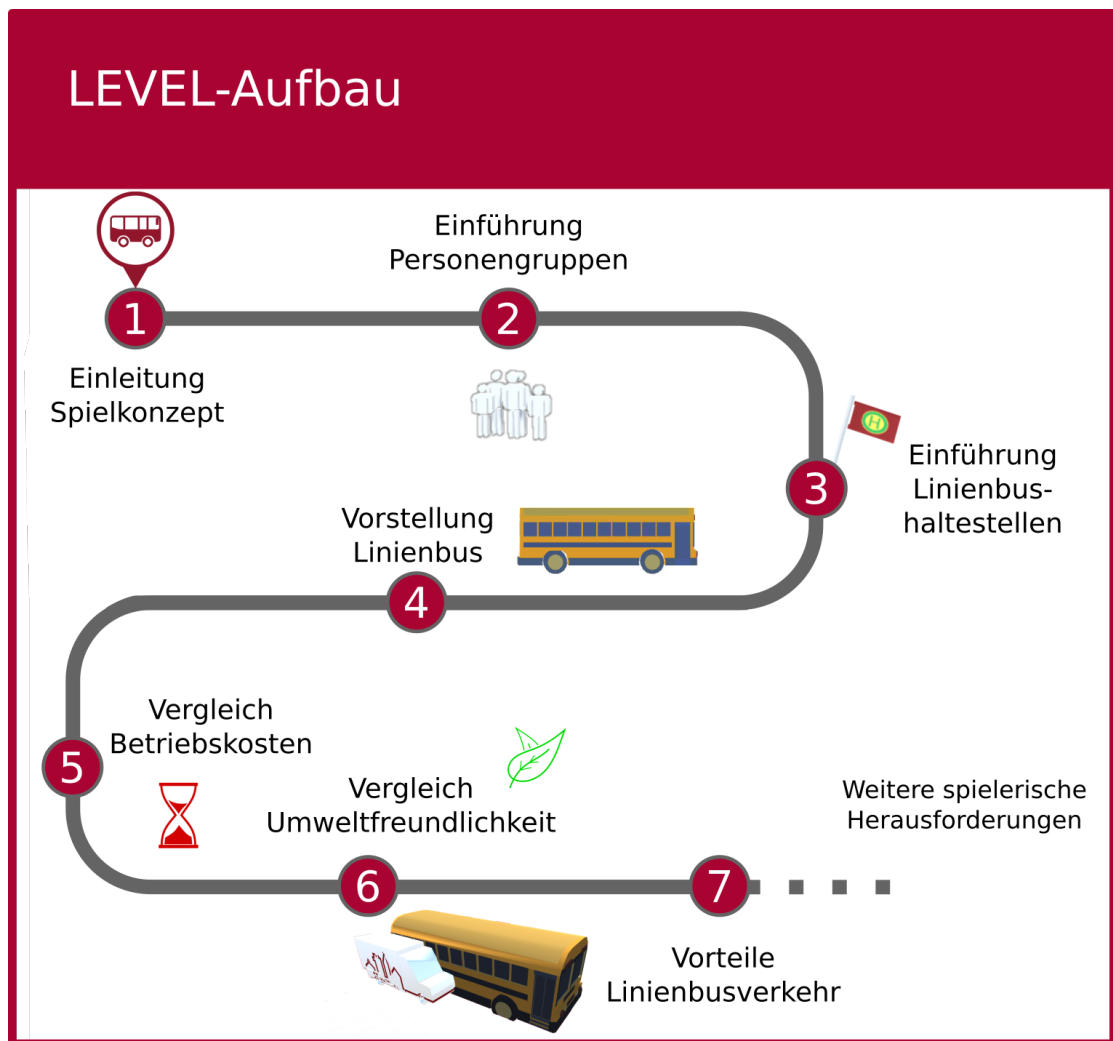
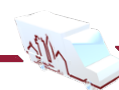


Abbildung 4.10: Planung der Spiellevel



5 Diskussion

Die Diskussion in dieser Arbeit besteht aus drei Teilen. Der erste Teil reflektiert die Ergebnisse der einzelnen Evaluationen, der zweite Teil beschreibt die Restriktionen bei der Benutzung des Vorgehensmodells und der Evaluationen. Der letzte Teil gibt schließlich einen Ausblick auf mögliche weitere Evaluationen und mögliche Weiterentwicklungen des Serious Games.

5.1 Reflektion der Ergebnisse

5.1.1 Allgemeine Beobachtungen bei den Evaluationen

Im Rahmen der ersten Usability-Evaluation wurde die Thinking-Aloud-Methode und die heuristische Methode direkt nacheinander durchgeführt. Damit sollte eine Vergleichbarkeit zwischen den beiden Methoden geschaffen werden. Das Diagramm in Abbildung 5.1 stellt einen direkten Vergleich der Problem-Nennungen dar. Das Diagramm zeigt, dass die Anzahl und die Nennungen der Probleme bei beiden Evaluationen ähnlich ist. Beide Evaluationsmethoden haben 22 gefundene Probleme als Ergebnis. Die Vorbereitungen der beiden Methoden haben einen vergleichbaren Umfang. Für beiden Methoden müssen Einweisungen vorbereitet werden. Für Thinking Aloud ist ein Aufnahmegerät erforderlich, für die heuristischen Evaluationen muss eine Tabelle vorbereitet werden. Die Durchführungen und Auswertungen waren allerdings unterschiedlich. Die Thinking Aloud-Methode erlaubt eine einfachere Durchführung, die Heuristische Methode bietet hingegen eine einfachere Auswertung. Die Unterschiede bei den Durchführungen und Auswertungen beschreiben die Kapitel 5.1.2 und 5.1.3.

Bei beiden Evaluationen wurden nur eine geringe Teilnehmerzahl verwendet. Wie in Kapitel 3.3.2 beschrieben soll eine Studie mit 5 Teilnehmern bis zu 85% aller Usability-Probleme aufdecken. Demnach reicht eine geringe Teilnehmeranzahl bei einer Studie aus. Ob tatsächlich in den Evaluationen mehr als 85% aller Usability-Probleme gefunden wurde ist schwer zu

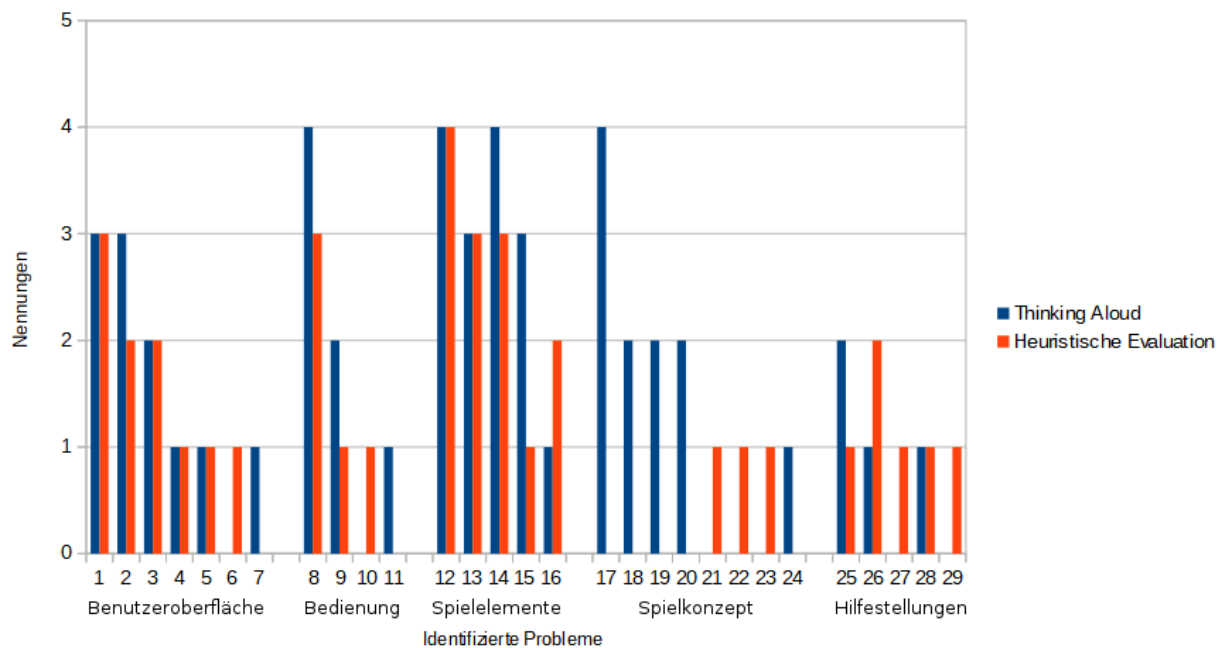


Abbildung 5.1: Nennungen der Probleme bei den Evaluationen

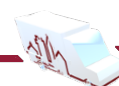
beweisen. Es konnte allerdings beobachtet werden, dass sich nach nur wenigen Teilnehmern die Nennungen der Probleme deutlich wiederholt haben.

Des Weiteren hat sich gezeigt, dass durch unterschiedliche Expertengruppen (z.B. Verkehrspsychologie- und Softwareentwicklungsexperten) unterschiedliche Ergebnisse entstehen können. Die schwerwiegendsten Probleme wurden von allen Experten entdeckt, manche Probleme wurden aber nur von einigen wenigen Testern gefunden und genannt.

5.1.2 Heuristische Evaluation

Die Durchführung der Heuristischen Evaluation ist im direkten Vergleich aufwendiger als die Durchführung der Thinking Aloud-Test. Für die Tester erfordert die Methode mehr Erklärungsbedarf, da die Heuristiken und deren Einordnung zu Beginn der Durchführung erklärt werden müssen.

Bei der Durchführung ist aufgefallen, dass zu viele Heuristiken unübersichtlich und verwirrend für die Tester sein können. Die Tester können bei zu viele Heuristiken den Überblick verlieren und müssen häufiger die genaue Bedeutung nachlesen, was den Dokumentationsfluss der Probleme stört. Je länger die Dokumentation dauert, desto größer ist die Gefahr des Vergessens von Problemen.



Bei dem direkten Vergleich der Ergebnisse fällt auf, dass die Anzahl und Nennungen der gefundenen Probleme bei der heuristischen Evaluation ähnlich ist, wie bei der Thinking Aloud-Test. Bei vereinzelt Problemen ist die Anzahl der Nennungen geringer, da die Tester die Probleme nur indirekt dokumentierten und teilweise die Probleme bei der Auflistung vergessen haben. Die Zuordnung der Heuristiken ist sehr unterschiedlich bei den verschiedenen Testern.

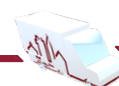
Der erste Schritt der Auswertung, die Digitalisierung der Probleme, ist bei der Heuristischen Evaluation einfacher, da die Ergebnisse bereits ausformuliert sind. Die meisten Probleme können direkt übernommen werden, Doppelnennungen treten nur selten auf. Die Heuristiken und die subjektiven Prioritäten helfen bei der Einordnung und Festlegung der endgültigen Priorität für das Projekt.

5.1.3 Thinking Aloud

Die Thinking Aloud - Test gewährt eine schnelle und einfache Durchführung. Sie sorgt für viele Ergebnisse in einer relativ kurzen Testzeit.

Die Vorbereitung und Durchführung haben sich sowohl für die Testenden als auch für die Evaluierenden als kurz und einfach erwiesen. Für die Tester war es einfach, da die Durchführung nur wenige Anweisungen erfordert, der Tester keine separate Dokumentationspflicht hat und die Aufnahme der Ergebnisse direkt durch eine Audioaufnahme erfolgt. Für den Evaluierenden ist die Vorbereitung einfach, da für die Durchführung nur das Testobjekt und ein Aufnahmegerät benötigt wird. Es werden keine langen Einweisungsdokumente, keine aufwendigen Aufgaben und keine Heuristiken benötigt.

Allerdings ist die Auswertung im Vergleich zur heuristische Evaluation aufwendiger. Für die Auswertung ist ein erneutes Anhören der Aufnahme und eine Analysieren notwendig. Das Einordnen der gefundenen Probleme in Kategorien und die Priorisierung muss ebenfalls der Evaluierende übernehmen. Für die Priorisierung kann die Mimik, Gestik und der Sprachkontext Hinweise geben, allerdings sind diese Hinweise nicht eindeutig.



5.2 Restriktionen

Restriktionen zur Benutzung des Vorgehensmodells

Das gewählte Vorgehensmodell legt einen Schwerpunkt auf die Evaluation und Produktion der Software. Es geht von einer iterativen Verbesserung des Produkts aus. Die Güte des Produkts ist allerdings nicht direkt messbar. Die Iterationsschleifen für B.u.S wurden schon früh im Projekt geplant. In der industriellen Entwicklung ist der Zeitplan nicht immer gut planbar, da die Zeit für die Umsetzung von bestimmten Features selten genau bekannt ist und die Güte schlecht planbar ist. Deshalb sollte bei jedem iterativen Projekt Zeit für zusätzliche Iterationen eingeplant werden.

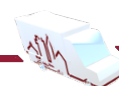
Das Vorgehensmodell für B.u.S. endet bei der Übergabe des Projekts vor dem Verteilen der Software. Das Verteilen und die Wartung der Software bildet das Vorgehensmodell nicht ab und muss getrennt geplant werden. Vor dem Verteilen der Software ist es eventuell noch sinnvoll eine weitere Evaluation mit einer unabhängigen Zielgruppen durchzuführen.

Des Weiteren ist das Vorgehensmodell für ein Projekt mit nur einem Entwicklerteam ausgelegt. Bei größeren Entwicklungsumfängen kann ein Projekt in Unterprojekten unterteilt werden. Dies bildet das Modell allerdings nicht ab. Bei mehreren Unterprojekten können die Iterationen parallel ablaufen.

Restriktionen zur Verwendung der Usability-Evaluationen

Die benutzten Usability-Evaluationen haben erfolgreich Usability-Probleme in dem Serious Game aufgedeckt. Eine Post-Evaluation war zusätzlich notwendig, um zu zeigen, dass sich die Qualität des Serious Games verbessert hat. Die Anzahl der nicht gefundenen Usability-Probleme kann allerdings nicht bestimmt werden. Dank der guten Motivation und Mitarbeit der Tester wurden viele Probleme gefunden. Eine schlechte Motivation des Tester kann sich gegebenenfalls auf die Ergebnis einer Evaluation auswirken.

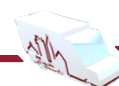
Durch die begrenzte Anzahl an Testern kann nicht bestimmt werden, ob B.u.S. für die breit definierte Zielgruppe geeignet ist. Dieses muss durch weitere Evaluationen nachgewiesen werden, wie in Kapitel 5.3 beschrieben wird.



5.3 Ausblick und Empfehlungen

Mit dem Serious Game B.u.S. wurde ein didaktisches Instrument geschaffen um Wissen im Bereich von modernen Bedarfsverkehren zu vermitteln. Diese Arbeit hat lediglich das Instrument dafür geschaffen, die Effektivität und Wirksamkeit des Instruments muss durch weitere Studien verifiziert werden. Das benutzte Vorgehensmodell sieht bewusst nur Evaluationen vor und beschreibt keine Teilnehmergruppen. Für die Entwicklung eines Konzepts mit einer prototypischen Implementierung ist die Evaluierung mit einer kleinen Expertengruppen ausreichend. Für weitere Evaluationen und Softwaretests ist es sinnvoll eine größere Teilnehmergruppe zu berücksichtigen. Die nächste Teilnehmergruppe sollte an der Zielgruppe des Serious Games ausgerichtet sein und eine größere Anzahl an Testern enthalten. Vor einer möglichen Verteilung der Software sollten die Softwaretests zur Qualitätssicherung ebenfalls ausgeweitet werden. Nach der Verteilung ist eine Wartung der Software notwendig.

Eine Fortführung des Projekts kann auch eine Verbesserung des Serious Games beinhalten. Zukünftige Versionen sollten mehr Level beinhalten um die Nutzer stärker zu binden. Der Inhalt der Level von B.u.S. kann durch zusätzliche Interaktionen mit Spielelementen und durch die Darstellung weiterer Verkehrsmittel aufgewertet werden. Das Serious Game ist als Stand-Alone-Software geplant. Für die Implementierung einer internen Rangliste oder für weitere integrierte Evaluationen ist eine Anbindung an eine Back-End-Software mit einer Datenbank sinnvoll sein.



A Anhang

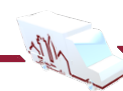
Inhalt der beigelegten DVD:

- A - Fokusgruppen-Präsentation
- B - Evaluationsbögen
- C - Audioaufnahmen der Evaluationen
- D - Serious Game als Unity-Projekt
- E - Serious Game als Android- und PC-Version
- F - Masterarbeit als PDF-Datei

Literaturverzeichnis

- Abt, C. C. (1970). *Serious Games*. New York: Viking Compass.
- Aleem, S., Fernando Capretz, L. & Ahmed, F. (2016). „Game Development Software Engineering Process Life Cycle: A Systematic Review“. In: *Journal of Software Engineering Research and Development*, 4(6):1–30.
- Beck, K. (2001). *Manifesto for Agile Software Development*. URL: <http://agilemanifesto.org> (besucht am 07.06.2018).
- BMVBS (2009). *Handbuch zur Planung flexibler Bedienungsformen im ÖPNV*. Bundesministerium für Verkehr, Bau und Stadtentwicklung (BMVBS). Bonn.
- Boehm, B. (1988). „A spiral model of software development and enhancement“. In: *IEEE Computer*, S. 61 –72.
- Brandt-Pook, H. & Kollmeier, R. (2008). *Softwareentwicklung kompakt und verständlich: Wie Softwaresysteme entstehen*. Wiesbaden: Vieweg + Teubner.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P. & Stal, M. (1998). *Pattern-orientierte Software-Architektur*. Addison-Wesley.
- Charters, E. (2003). „The Use of Think-aloud Methods in Qualitative Research - An Introduction to Think-aloud Methods“. In: S. 68 –82.
- Deppe, M. (2013). „Cities in Motion 2 im Test - Das bessere SimCity?“ In: *GameStar*.
- Eckermann, J. (1836). *Gespräche mit Goethe in den letzten Jahren seines Lebens*. Frankfurt a. M.: Insel Verlag.
- Epping, T. (2011). *Kanban für die Softwareentwicklung*. Berlin, Heidelberg: Springer Vieweg.
- Fullerton, T. (2008). *Game Design Workshop - A Playcentric Approach to Creating Innovative Games*. Burlington, USA: Elsevier.
- GaPSlabs (o.D.). *Website ProtoWorld*. URL: <http://www.gapslabs.org/protoworld/> (besucht am 28.04.2018).
- Graf, M. (2008). „Ernsthafte Spiele - »Serious Games« erobern die Welt. Wir klären auf!“ In: *GameStar*.

- Hunicke, R., LeBlanc, M. & Zubek, R. (2004). *MDA: A Formal Approach to Game Design and Game Research*.
- International Organization for Standardization (2011). *DIN EN ISO 9241-11 Anforderungen an die Gebrauchstauglichkeit*.
- Kleuker, S. (2013). *Grundkurs Software-Engineering mit UML*. Wiesbaden.
- Klix, F. & Spada, H. (1998). „C/II/6 Enzyklopädie der Psychologie“. In: Hogrefe. Kap. Wissensvermittlung: Ansätze zur Förderung des Wissenserwerbs aus Wissen, Kapitel: 14, S. 457 –500.
- König, A., J., G., Wegener, J. & A., P. (2017). „Serious Games: A playful approach to reduce usage barriers of innovative public transport systems“. In: *European Transport Conference*.
- Krueger, R. A. (1998). *Analyzing & Reporting Focus Group Results*. SAGE.
- Nash, A. (o.D.). *BusMeister Project-Website*. URL: <http://crowdsourced-transport.com/our-projects/busmeister-project/>.
- Nielsen, J. (1995a). *10 Usability Heuristics for User Interface Design*. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (besucht am 07.06.2018).
- Nielsen, J. (1995b). *Severity Ratings for Usability Problems*. URL: <https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/> (besucht am 07.06.2018).
- Nielsen, J. (2000). *Why You Only Need to Test with 5 Users*. URL: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/> (besucht am 07.06.2018).
- Nielsen, J. (2012). *Thinking Aloud: The 1 Usability Tool*. URL: <https://www.nngroup.com/articles/thinking-aloud-the-1-usability-tool/> (besucht am 07.06.2018).
- Pagulayan, R. J., Kecker, K., Fuller, T., Wixon, D., Romero, R. L. & Gunn, D. V. (2007). *User-centered Design in Games*. Human-Computer Interaction Handbook. URL: http://msgamesresearch.com/Documents/HCI_Handbook_Chapter.pdf (besucht am 07.06.2018).
- Pelz, A. (2017). „User Experience Design für ein Serious Game auf mobilen Plattformen“. Magisterarb. Wernigerode: Hochschule Harz.
- Prensky, M. (2007). *Digital Game-based Learning*. St. Paul, Ma.: Paragon House.
- Prühs, N. (2014). *Component-Based Entity Systems in Spielen*. URL: <https://www.heise.de/developer/artikel/Component-Based-Entity-Systems-in-Spielen-2262126.html?seite=all> (besucht am 07.06.2018).
- Ramadan, R. & Widayani, Y. (2013). „Game Development Life Cycle Guidelines“. In: *ICAC-SIS*.



- Rath, R. (2017). *We Asked a Transit Planner How to Up Our 'Mini Metro' Game*. URL: https://waypoint.vice.com/en_us/article/8qgw4p/we-asked-a-transit-planner-how-to-up-our-mini-metro-game (besucht am 07.06.2018).
- Rehfeld, G. (2013). *Game Design und Produktion - Grundlagen, Anwendungen und Beispiele*. Carl Hanser Verlag.
- Richter, M. & Flückinger, M. D. (2016). *Usability und UX kompakt*. Berlin, Heidelberg: Springer Vieweg.
- Rosenzweig, E. (2015). *Successful User Experience- Strategies and Roadmaps*. Morgan Kaufmann.
- Salzburg Research Forschungsgesellschaft (o.D.). *Fokusgruppe für Geschäftsmodell-Entwicklung*. URL: <https://methodenpool.salzburgresearch.at/methode/fokusgruppe/> (besucht am 07.06.2018).
- Sawyer, B. & Rejeski, D. (2002). „Serious Games: Improving Public Policy Through Game-based Learning and Simulation“. In:
- Schulz, M., Mack, B. & Renn, O. (2012). *Fokusgruppen in der empirischen Sozialwissenschaft - Von der Konzeption bis zur Auswertung*. Springer VS.
- Spillner, A. & Linz, T. (2012). *Basiswissen Softwaretest : Aus- und Weiterbildung zum Certified Tester*. 5. Heidelberg: dpunkt. Verlag.
- Stadt Schorndorf (2017). *Website Reallabor Schorndorf*. URL: <http://www.reallabor-schorndorf.de/> (besucht am 07.06.2018).
- Strahringer, S. & Leyh, C. (2017). *Gamification und Serious Games*. Wiesbaden: Springer Vieweg.
- Terhart, E. (1997). *Lehr-Lern-Methoden: Eine Einführung in Probleme der methodischen Organisation von Lehren und Lernen*. Juventa.
- Unity Technologies (o.D.). *Website Unity*. URL: <https://unity3d.com/de> (besucht am 07.06.2018).
- Wegener, J. (2017). „Konzeptentwicklung eines KI-Agenten für ein Serious Game“. Masterarb. Wernigerode: Hochschule Harz.
- Zyda, M. (2005). „From visual simulation to virtual reality to games“. In: *IEEE Computer Society*, S. 25–32.

